CLI Book 1: Cisco ASA Series General Operations CLI Configuration Guide, 9.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

PREFACE

PART I

CHAPTER 1

PART II

CHAPTER 7

About This Guide

The following topics explain how to use this guide.

Document Objectives

The purpose of this guide is to help you configure general operations for the Cisco ASA series using the command-line interface. This guide does not cover every feature, but describes only the most common configuration scenarios.

You can also configure and monitor the ASA by using the Adaptive Security Device Manager (ASDM), a web-based GUI application. ASDM includes configuration wizards to guide you through some common configuration scenarios, and online help for less common scenarios.

Throughout this guide, the term “ASA” applies generically to supported models, unless specified otherwise.

Related Documentation

For more information, see *Navigating the Cisco ASA Series Documentation* at http://www.cisco.com/go/asadocs.

Document Conventions

This document adheres to the following text, display, and alert conventions.

Text Conventions

Convention	Indication
boldface	Commands, keywords, button labels, field names, and user-entered text appear in **boldface**. For menu-based commands, the full path to the command is shown.

Convention	Indication
italic	Variables, for which you supply values, are presented in an *italic* typeface. Italic type is also used for document titles, and for general emphasis.
`monospace`	Terminal sessions and information that the system displays appear in `monospace` type.
{x | y | z}	Required alternative keywords are grouped in braces and separated by vertical bars.
[ ]	Elements in square brackets are optional.
[x | y | z]	Optional alternative keywords are grouped in square brackets and separated by vertical bars.
[ ]	Default responses to system prompts are also in square brackets.
< >	Non-printing characters such as passwords are in angle brackets.
!, #	An exclamation point (!) or a number sign (#) at the beginning of a line of code indicates a comment line.

Reader Alerts

This document uses the following for reader alerts:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem.*

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time.* You can save time by performing the action described in the paragraph.

Warning **Means *reader be warned.* In this situation, you might perform an action that could result in bodily injury.**

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

PART I

Getting Started with the ASA

CHAPTER 1

Introduction to the Cisco ASA

The Cisco ASA provides advanced stateful firewall and VPN concentrator functionality in one device as well as integrated services with add-on modules. The ASA includes many advanced features, such as multiple security contexts (similar to virtualized firewalls), clustering (combining multiple firewalls into a single firewall), transparent (Layer 2) firewall or routed (Layer 3) firewall operation, advanced inspection engines, IPsec VPN, SSL VPN, and clientless SSL VPN support, and many more features.

Hardware and Software Compatibility

For a complete list of supported hardware and software, see Cisco ASA Compatibility.

VPN Compatibility

See Supported VPN Platforms, Cisco ASA Series.

New Features

This section lists new features for each release.

Note New, changed, and deprecated syslog messages are listed in the syslog message guide.

New Features in ASA 9.4(4.5)

Released: April 3, 2017

Note Verion 9.4(4) was removed from Cisco.com due to bug CSCvd78303.

There are no new features in this release.

New Features in ASA 9.4(3)

Released: April 25, 2016

Feature	Description
Firewall Features	
Connection holddown timeout for route convergence	You can now configure how long the system should maintain a connection when the route used by the connection no longer exists or is inactive. If the route does not become active within this holddown period, the connection is freed. You can reduce the holddown timer to make route convergence happen more quickly. However, the 15 second default is appropriate for most networks to prevent route flapping. We added the following command: **timeout conn-holddown**
Remote Access Features	
Configurable SSH encryption and HMAC algorithm.	Users can select cipher modes when doing SSH encryption management and can configure HMAC and encryption for varying key exchange algorithms. We introduced the following commands: **ssh cipher encryption, ssh cipher integrity**. *Also available in 9.1(7).*
HTTP redirect support for IPv6	When you enable HTTP redirect to HTTPS for ASDM access or clientless SSL VPN, you can now redirect traffic sent an to IPv6 address. We added functionality to the following command: **http redirect** *Also available in 9.1(7).*
Monitoring Features	
SNMP engineID sync for Failover	In a failover pair, the SNMP engineIDs of the paired ASAs are synced on both units. Three sets of engineIDs are maintained per ASA—synced engineID, native engineID and remote engineID. An SNMPv3 user can also specify the engineID of the ASA when creating a profile to preserve localized **snmp-server user** authentication and privacy options. If a user does not specify the native engineID, the **show running config** output will show two engineIDs per user. We modified the following command: **snmp-server user**

Feature	Description
show tech support enhancements	The **show tech support** command now: • Includes **dir all-filesystems** output—This output can be helpful in the following cases: • SSL VPN configuration: check if the required resources are on the ASA • Crash: check for the date timestamp and presence of a crash file • Removes the **show kernel cgroup-controller detail** output—This command output will remain in the output of **show tech-support detail**. We modified the following command: **show tech support** *Also available in 9.1(7).*
Support for the cempMemPoolTable in the CISCO-ENHANCED-MEMPOOL-MIB	The cempMemPoolTable of the CISCO-ENHANCED-MEMPOOL-MIB is now supported. This is a table of memory pool monitoring entries for all physical entities on a managed system. **Note** The CISCO-ENHANCED-MEMPOOL-MIB uses 64-bit counters and supports reporting of memory on platforms with more than 4GB of RAM. We did not add or modify any commands. *Also available in 9.1(7).*

New Features in ASA 9.4(2.145)

Released: November 13, 2015

There are no new features in this release.

Note This release supports only the Firepower 9300 ASA security module.

New Features in ASA 9.4(2)

Released: September 24, 2015

There are no new features in this release.

Note ASAv 9.4(1.200) features are not included in this release.

Note This version does not support the ISA 3000.

New Features in ASA 9.4(1.225)

Released: September 17, 2015

Note This release supports only the Cisco ISA 3000.

Feature	Description
Platform Features	
Cisco ISA 3000 Support	The Cisco ISA 3000 is a DIN Rail mounted, ruggedized, industrial security appliance. It is low-power, fan-less, with Gigabit Ethernet and a dedicated management port. This model comes with the ASA Firepower module pre-installed. Special features for this model include a customized transparent mode default configuration, as well as a hardware bypass function to allow traffic to continue flowing through the appliance when there is a loss of power. We introduced the following commands: **hardware-bypass, hardware-bypass manual, hardware-bypass boot-delay, show hardware-bypass** *This feature is not available in Version 9.5(1).*

New Features in ASA 9.4(1.152)

Released: July 13, 2015

Note This release supports only the ASA on the Firepower 9300.

Feature	Description
Platform Features	
ASA security module on the Firepower 9300	We introduced the ASA security module on the Firepower 9300. **Note** Firepower Chassis Manager 1.1.1 does not support any VPN features (site-to-site or remote access) for the ASA security module on the Firepower 9300.
High Availability Features	
Intra-chassis ASA Clustering for the Firepower 9300	You can cluster up to 3 security modules within the Firepower 9300 chassis. All modules in the chassis must belong to the cluster. We introduced the following commands: **cluster replication delay, debug service-module, management-only individual, show cluster chassis**
Licensing Features	

Feature	Description
Cisco Smart Software Licensing for the ASA on the Firepower 9300	We introduced Smart Software Licensing for the ASA on the Firepower 9300. We introduced the following commands: **feature strong-encryption, feature mobile-sp, feature context**

New Features in ASAv 9.4(1.200)

Released: May 12, 2015

Note This release supports only the ASAv.

Feature	Description
Platform Features	
ASAv on VMware no longer requires vCenter support	You can now install the ASAv on VMware without vCenter using the vSphere client or the OVFTool using a Day 0 configuration.
ASAv on Amazon Web Services (AWS)	You can now use the ASAv with Amazon Web Services (AWS) and the Day 0 configuration. **Note** Amazon Web Services only supports models ASAv10 and ASAv30.

New Features in ASA 9.4(1)

Released: March 30, 2015

Feature	Description
Platform Features	
ASA 5506W-X, ASA 5506H-X, ASA 5508-X, ASA 5516-X	We introduced the ASA 5506W-X with wireless access point, hardened ASA 5506H-X, ASA 5508-X, and ASA 5516-X models. We introduced the following command: **hw-module module wlan recover image**, **hw-module module wlan recover image**.
Certification Features	

Feature	Description
Department of Defense Unified Capabilities Requirements (UCR) 2013 Certification	The ASA was updated to comply with the DoD UCR 2013 requirements. See the rows in this table for the following features that were added for this certification: • Periodic certificate authentication • Certificate expiration alerts • Enforcement of the basic constraints CA flag • ASDM Username From Certificate Configuration • ASDM management authorization • IKEv2 invalid selectors notification configuration • IKEv2 pre-shared key in Hex
FIPS 140-2 Certification compliance updates	When you enable FIPS mode on the ASA, additional restrictions are put in place for the ASA to be FIPS 140-2 compliant. Restrictions include: • RSA and DH Key Size Restrictions—Only RSA and DH keys 2K (2048 bits) or larger are allowed. For DH, this means groups 1 (768 bit), 2 (1024 bit), and 5 (1536 bit) are not allowed. **Note** The key size restrictions disable use of IKEv1 with FIPS. • Restrictions on the Hash Algorithm for Digital Signatures—Only SHA256 or better is allowed. • SSH Cipher Restrictions—Allowed ciphers: aes128-cbc or aes256-cbc. MACs: SHA1 To see the FIPS certification status for the ASA, see: http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140InProcess.pdf This PDF is updated weekly. See the Computer Security Division Computer Security Resource Center site for more information: http://csrc.nist.gov/groups/STM/cmvp/inprocess.html We modified the following command: **fips enable**
Firewall Features	
Improved SIP inspection performance on multiple core ASAs.	If you have multiple SIP signaling flows going through an ASA with multiple cores, SIP inspection performance has been improved. However, you will not see improved performance if you are using a TLS, phone, or IME proxy. We did not modify any commands.
SIP inspection support for Phone Proxy and UC-IME Proxy was removed.	You can no longer use Phone Proxy or UC-IME Proxy when configuring SIP inspection. Use TLS Proxy to inspect encrypted traffic. We removed the following commands: **phone-proxy**, **uc-ime**. We removed the **phone-proxy** and **uc-ime** keywords from the **inspect sip** command.

Feature	Description
DCERPC inspection support for ISystemMapper UUID message RemoteGetClassObject opnum3.	The ASA started supporting non-EPM DCERPC messages in release 8.3, supporting the ISystemMapper UUID message RemoteCreateInstance opnum4. This change extends support to the RemoteGetClassObject opnum3 message. We did not modify any commands.
Unlimited SNMP server trap hosts per context	The ASA supports an unlimited number of SNMP server trap hosts per context. The **show snmp-server host** command output displays only the active hosts that are polling the ASA, as well as the statically configured hosts. We modified the following command: **show snmp-server host**.
VXLAN packet inspection	The ASA can inspect the VXLAN header to enforce compliance with the standard format. We introduced the following command: **inspect vxlan**.
DHCP monitoring for IPv6	You can now monitor DHCP statistics and DHCP bindings for IPv6.
ESMTP inspection change in default behavior for TLS sessions.	The default for ESMTP inspection was changed to allow TLS sessions, which are not inspected. However, this default applies to new or reimaged systems. If you upgrade a system that includes **no allow-tls**, the command is not changed. The change in default behavior was also made in these older versions: 8.4(7.25), 8.5(1.23), 8.6(1.16), 8.7(1.15), 9.0(4.28), 9.1(6.1), 9.2(3.2) 9.3(1.2), 9.3(2.2).
High Availability Features	
Blocking syslog generation on a standby ASA	You can now block specific syslogs from being generated on a standby unit. We introduced the following command: **no logging message** *syslog-id* **standby**.
Enable and disable ASA cluster health monitoring per interface	You can now enable or disable health monitoring per interface. Health monitoring is enabled by default on all port-channel, redundant, and single physical interfaces. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored. You might want to disable health monitoring of non-essential interfaces, for example, the management interface. We introduced the following command: **health-check monitor-interface.**
ASA clustering support for DHCP relay	You can now configure DHCP relay on the ASA cluster. Client DHCP requests are load-balanced to the cluster members using a hash of the client MAC address. DHCP client and server functions are still not supported. We introduced the following command: **debug cluster dhcp-relay**
SIP inspection support in ASA clustering	You can now configure SIP inspection on the ASA cluster. A control flow can be created on any unit (due to load balancing), but its child data flows must reside on the same unit. TLS Proxy configuration is not supported. We introduced the following command: **show cluster service-policy**
Routing Features	

Feature	Description
Policy Based Routing	Policy Based Routing (PBR) is a mechanism by which traffic is routed through specific paths with a specified QoS using ACLs. ACLs let traffic be classified based on the content of the packet's Layer 3 and Layer 4 headers. This solution lets administrators provide QoS to differentiated traffic, distribute interactive and batch traffic among low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths, and allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections. We introduced the following commands: **set ip next-hop verify-availability, set ip next-hop, set ip next-hop recursive, set interface, set ip default next-hop, set default interface, set ip df, set ip dscp, policy-route route-map, show policy-route, debug policy-route**
Interface Features	
VXLAN support	VXLAN support was added, including VXLAN tunnel endpoint (VTEP) support. You can define one VTEP source interface per ASA or security context. We introduced the following commands: **debug vxlan, default-mcast-group, encapsulation vxlan, inspect vxlan, interface vni, mcast-group, nve, nve-only, peer ip, segment-id, show arp vtep-mapping, show interface vni, show mac-address-table vtep-mapping, show nve, show vni vlan-mapping, source-interface, vtep-nve, vxlan port**
Monitoring Features	
Memory tracking for the EEM	We have added a new debugging feature to log memory allocations and memory usage, and to respond to memory logging wrap events. We introduced or modified the following commands: **memory logging, show memory logging, show memory logging include, event memory-logging-wrap**
Troubleshooting crashes	The **show tech-support** command output and **show crashinfo** command output includes the most recent 50 lines of generated syslogs. Note that you must enable the **logging buffer** command to enable these results to appear.
Remote Access Features	

Feature	Description
Support for ECDHE-ECDSA ciphers	TLSv1.2 added support for the following ciphers: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • DHE-RSA-AES256-GCM-SHA384 • AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • DHE-RSA-AES128-GCM-SHA256 • RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 **Note** ECDSA and DHE ciphers are the highest priority. We introduced the following command: **ssl ecdh-group**.
Clientless SSL VPN session cookie access restriction	You can now prevent a Clientless SSL VPN session cookie from being accessed by a third party through a client-side script such as Javascript. **Note** Use this feature only if Cisco TAC advises you to do so. Enabling this command presents a security risk because the following Clientless SSL VPN features will not work without any warning. • Java plug-ins • Java rewriter • Port forwarding • File browser • Sharepoint features that require desktop applications (for example, MS Office applications) • AnyConnect Web launch • Citrix Receiver, XenDesktop, and Xenon • Other non-browser-based and browser plugin-based applications We introduced the following command: **http-only-cookie**. *This feature is also in 9.2(3).*

Feature	Description
Virtual desktop access control using security group tagging	The ASA now supports security group tagging-based policy control for Clientless SSL remote access to internal applications and websites. This feature uses Citrix's virtual desktop infrastructure (VDI) with XenDesktop as the delivery controller and the ASA's content transformation engine. See the following Citrix product documentation for more information: • Policies for XenDesktop and XenApp: http://support.citrix.com/proddocs/topic/infocenter/ic-how-to-use.html • Managing policies in XenDesktop 7: http://support.citrix.com/proddocs/topic/xendesktop-7/cds-policies-wrapper-rho.html • Using group policy editor for XenDesktop 7 policies: http://support.citrix.com/proddocs/topic/xendesktop-7/cds-policies-use-gpmc.html
OWA 2013 feature support has been added for Clientless SSL VPN	Clientless SSL VPN supports the new features in OWA 2013 except for the following: • Support for tablets and smartphones • Offline mode • Active Directory Federation Services (AD FS) 2.0. The ASA and AD FS 2.0 can't negotiate encryption protocols. We did not modify any commands.
Citrix XenDesktop 7.5 and StoreFront 2.5 support has been added for Clientless SSL VPN	Clientless SSL VPN supports the access of XenDesktop 7.5 and StoreFront 2.5. See http://support.citrix.com/proddocs/topic/xenapp-xendesktop-75/cds-75-about-whats-new.html for the full list of XenDesktop 7.5 features, and for more details. See http://support.citrix.com/proddocs/topic/dws-storefront-25/dws-about.html for the full list of StoreFront 2.5 features, and for more details. We did not modify any commands.
Periodic certificate authentication	When you enable periodic certificate authentication, the ASA stores certificate chains received from VPN clients and re-authenticates them periodically. We introduced or modified the following commands: **periodic-authentication certificate, revocation-check, show vpn-sessiondb**
Certificate expiration alerts	The ASA checks all CA and ID certificates in the trust points for expiration once every 24 hours. If a certificate is nearing expiration, a syslog will be issued as an alert. You can configure the reminder and recurrence intervals. By default, reminders will start at 60 days prior to expiration and recur every 7 days. We introduced or modified the following commands: **crypto ca alerts expiration**

Feature	Description
Enforcement of the basic constraints CA flag	Certificates without the CA flag now cannot be installed on the ASA as CA certificates by default. The basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate. You can configure the ASA to allow installation of these certificates if desired. We introduced the following command: **ca-check**
IKEv2 invalid selectors notification configuration	Currently, if the ASA receives an inbound packet on an SA, and the packet's header fields are not consistent with the selectors for the SA, then the ASA discards the packet. You can now enable or disable sending an IKEv2 notification to the peer. Sending this notification is disabled by default. **Note** This feature is supported with AnyConnect 3.1.06060 and later. We introduced the following command: **crypto ikev2 notify invalid-selectors**
IKEv2 pre-shared key in Hex	You can now configure the IKEv2 pre-shared keys in hex. We introduced the following command: **ikev2 local-authentication pre-shared-key hex**, **ikev2 remote-authentication pre-shared-key hex**
Administrative Features	
ASDM management authorization	You can now configure management authorization separately for HTTP access vs. Telnet and SSH access. We introduced the following command: **aaa authorization http console**
ASDM Username From Certificate Configuration	When you enable ASDM certificate authentication (**http authentication-certificate**), you can configure how ASDM extracts the username from the certificate; you can also enable pre-filling the username at the login prompt. We introduced the following command: **http username-from-certificate**
terminal interactive command to enable or disable help when you enter **?** at the CLI	Normally, when you enter **?** at the ASA CLI, you see command help. To be able to enter **?** as text within a command (for example, to include a ? as part of a URL), you can disable interactive help using the **no terminal interactive** command. We introduced the following command: terminal interactive
REST API Features	
REST API Version 1.1	We added support for the REST API Version 1.1.
Support for token-based authentication (in addition to existing basic authentication)	Client can send log-in request to a specific URL; if successful, a token is returned (in response header). Client then uses this token (in a special request header) for sending additional API calls. The token is valid until explicitly invalidated, or the idle/session timeout is reached.

Feature	Description
Limited multiple-context support	The REST API agent can now be enabled in multi-context mode; the CLI commands can be issued only in system-context mode (same commands as single-context mode). Pass-through CLI API commands can be used to configure any context, as follows. `https://<asa_admin_context_ip>/api/cli?context=<context_name>` If the **context** parameter is not present, it is assumed that the request is directed to the **admin** context.
Advanced (granular) inspection	Granular inspection of these protocols is supported: • DNS over UDP • HTTP • ICMP • ICMP ERROR • RTSP • SIP • FTP • DCERPC • IP Options • NetBIOS Name Server over IP • SQL*Net

Firewall Functional Overview

Firewalls protect inside networks from unauthorized access by users on an outside network. A firewall can also protect inside networks from each other, for example, by keeping a human resources network separate from a user network. If you have network resources that need to be available to an outside user, such as a web or FTP server, you can place these resources on a separate network behind the firewall, called a *demilitarized zone* (DMZ). The firewall allows limited access to the DMZ, but because the DMZ only includes the public servers, an attack there only affects the servers and does not affect the other inside networks. You can also control when inside users access outside networks (for example, access to the Internet), by allowing only certain addresses out, by requiring authentication or authorization, or by coordinating with an external URL filtering server.

When discussing networks connected to a firewall, the *outside* network is in front of the firewall, the *inside* network is protected and behind the firewall, and a *DMZ*, while behind the firewall, allows limited access to outside users. Because the ASA lets you configure many interfaces with varied security policies, including many inside interfaces, many DMZs, and even many outside interfaces if desired, these terms are used in a general sense only.

Security Policy Overview

A security policy determines which traffic is allowed to pass through the firewall to access another network. By default, the ASA allows traffic to flow freely from an inside network (higher security level) to an outside network (lower security level). You can apply actions to traffic to customize the security policy.

Permitting or Denying Traffic with Access Rules

You can apply access rules to limit traffic from inside to outside, or allow traffic from outside to inside. For bridge group interfaces, you can also apply an EtherType access rule to allow non-IP traffic.

Applying NAT

Some of the benefits of NAT include the following:

- You can use private addresses on your inside networks. Private addresses are not routable on the Internet.
- NAT hides the local addresses from other networks, so attackers cannot learn the real address of a host.
- NAT can resolve IP routing problems by supporting overlapping IP addresses.

Protecting from IP Fragments

The ASA provides IP fragment protection. This feature performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the ASA. Fragments that fail the security check are dropped and logged. Virtual reassembly cannot be disabled.

Applying HTTP, HTTPS, or FTP Filtering

Although you can use access lists to prevent outbound access to specific websites or FTP servers, configuring and managing web usage this way is not practical because of the size and dynamic nature of the Internet.

You can configure Cloud Web Security on the ASA, or install an ASA module that provides URL and other filtering services, such as ASA CX or ASA FirePOWER. You can also use the ASA in conjunction with an external product such as the Cisco Web Security Appliance (WSA).

Applying Application Inspection

Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection.

Sending Traffic to Supported Hardware or Software Modules

Some ASA models allow you to configure software modules, or to insert hardware modules into the chassis, to provide advanced services. These modules provide additional traffic inspection and can block traffic based on your configured policies. You can send traffic to these modules to take advantage of these advanced services.

Applying QoS Policies

Some network traffic, such as voice and streaming video, cannot tolerate long latency times. QoS is a network feature that lets you give priority to these types of traffic. QoS refers to the capability of a network to provide better service to selected network traffic.

Applying Connection Limits and TCP Normalization

You can limit TCP and UDP connections and embryonic connections. Limiting the number of connections and embryonic connections protects you from a DoS attack. The ASA uses the embryonic limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination.

TCP normalization is a feature consisting of advanced TCP connection settings designed to drop packets that do not appear normal.

Enabling Threat Detection

You can configure scanning threat detection and basic threat detection, and also how to use statistics to analyze threats.

Basic threat detection detects activity that might be related to an attack, such as a DoS attack, and automatically sends a system log message.

A typical scanning attack consists of a host that tests the accessibility of every IP address in a subnet (by scanning through many hosts in the subnet or sweeping through many ports in a host or subnet). The scanning threat detection feature determines when a host is performing a scan. Unlike IPS scan detection that is based on traffic signatures, the ASA scanning threat detection feature maintains an extensive database that contains host statistics that can be analyzed for scanning activity.

The host database tracks suspicious activity such as connections with no return activity, access of closed service ports, vulnerable TCP behaviors such as non-random IPID, and many more behaviors.

You can configure the ASA to send system log messages about an attacker or you can automatically shun the host.

Firewall Mode Overview

The ASA runs in two different firewall modes:

- Routed
- Transparent

In routed mode, the ASA is considered to be a router hop in the network.

In transparent mode, the ASA acts like a “bump in the wire,” or a “stealth firewall,” and is not considered a router hop. The ASA connects to the same network on its inside and outside interfaces in a "bridge group".

You might use a transparent firewall to simplify your network configuration. Transparent mode is also useful if you want the firewall to be invisible to attackers. You can also use a transparent firewall for traffic that would otherwise be blocked in routed mode. For example, a transparent firewall can allow multicast streams using an EtherType access list.

Stateful Inspection Overview

All traffic that goes through the ASA is inspected using the Adaptive Security Algorithm and either allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks *every* packet against the filter, which can be a slow process.

Note The TCP state bypass feature allows you to customize the packet flow.

A stateful firewall like the ASA, however, takes into consideration the state of a packet:

- Is this a new connection?

  If it is a new connection, the ASA has to check the packet against access lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the “session management path,” and depending on the type of traffic, it might also pass through the “control plane path.”

  The session management path is responsible for the following tasks:

  - Performing the access list checks
  - Performing route lookups
  - Allocating NAT translations (xlates)
  - Establishing sessions in the “fast path”

  The ASA creates forward and reverse flows in the fast path for TCP traffic; the ASA also creates connection state information for connectionless protocols like UDP, ICMP (when you enable ICMP inspection), so that they can also use the fast path.

  Note For other IP protocols, like SCTP, the ASA does not create reverse path flows. As a result, ICMP error packets that refer to these connections are dropped.

  Some packets that require Layer 7 inspection (the packet payload must be inspected or altered) are passed on to the control plane path. Layer 7 inspection engines are required for protocols that have two or more channels: a data channel, which uses well-known port numbers, and a control channel, which uses different port numbers for each session. These protocols include FTP, H.323, and SNMP.

- Is this an established connection?

  If the connection is already established, the ASA does not need to re-check packets; most matching packets can go through the “fast” path in both directions. The fast path is responsible for the following tasks:

  - IP checksum verification
  - Session lookup
  - TCP sequence number check
  - NAT translations based on existing sessions

- Layer 3 and Layer 4 header adjustments

Data packets for protocols that require Layer 7 inspection can also go through the fast path.

Some established session packets must continue to go through the session management path or the control plane path. Packets that go through the session management path include HTTP packets that require inspection or content filtering. Packets that go through the control plane path include the control packets for protocols that require Layer 7 inspection.

VPN Functional Overview

A VPN is a secure connection across a TCP/IP network (such as the Internet) that appears as a private connection. This secure connection is called a tunnel. The ASA uses tunneling protocols to negotiate security parameters, create and manage tunnels, encapsulate packets, transmit or receive them through the tunnel, and unencapsulate them. The ASA functions as a bidirectional tunnel endpoint: it can receive plain packets, encapsulate them, and send them to the other end of the tunnel where they are unencapsulated and sent to their final destination. It can also receive encapsulated packets, unencapsulate them, and send them to their final destination. The ASA invokes various standard protocols to accomplish these functions.

The ASA performs the following functions:

- Establishes tunnels
- Negotiates tunnel parameters
- Authenticates users
- Assigns user addresses
- Encrypts and decrypts data
- Manages security keys
- Manages data transfer across the tunnel
- Manages data transfer inbound and outbound as a tunnel endpoint or router

The ASA invokes various standard protocols to accomplish these functions.

Security Context Overview

You can partition a single ASA into multiple virtual devices, known as security contexts. Each context is an independent device, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. Many features are supported in multiple context mode, including routing tables, firewall features, IPS, and management; however, some features are not supported. See the feature chapters for more information.

In multiple context mode, the ASA includes a configuration for each context that identifies the security policy, interfaces, and almost all the options you can configure on a standalone device. The system administrator adds and manages contexts by configuring them in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the ASA. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs into the admin context, then that user has system administrator rights and can access the system and all other contexts.

ASA Clustering Overview

ASA Clustering lets you group multiple ASAs together as a single logical device. A cluster provides all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices.

You perform all configuration (aside from the bootstrap configuration) on the control unit only; the configuration is then replicated to the member units.

Special and Legacy Services

For some services, documentation is located outside of the main configuration guides and online help.

Special Services Guides

Special services allow the ASA to interoperate with other Cisco products; for example, by providing a security proxy for phone services (Unified Communications), or by providing Botnet traffic filtering in conjunction with the dynamic database from the Cisco update server, or by providing WCCP services for the Cisco Web Security Appliance. Some of these special services are covered in separate guides:

- Cisco ASA Botnet Traffic Filter Guide
- Cisco ASA NetFlow Implementation Guide
- Cisco ASA Unified Communications Guide
- Cisco ASA WCCP Traffic Redirection Guide
- SNMP Version 3 Tools Implementation Guide

Legacy Services Guide

Legacy services are still supported on the ASA, however there may be better alternative services that you can use instead. Legacy services are covered in a separate guide:

Cisco ASA Legacy Feature Guide

This guide includes the following chapters:

- Configuring RIP
- AAA Rules for Network Access
- Using Protection Tools, which includes Preventing IP Spoofing (**ip verify reverse-path**), Configuring the Fragment Size (**fragment**), Blocking Unwanted Connections (**shun**), Configuring TCP Options (for ASDM), and Configuring IP Audit for Basic IPS Support (**ip audit**).
- Configuring Filtering Services

CHAPTER 2

Getting Started

This chapter describes how to get started with your Cisco ASA.

Access the Console for the Command-Line Interface

For initial configuration, access the CLI directly from the console port. Later, you can configure remote access using Telnet or SSH according to #unique_39. If your system is already in multiple context mode, then accessing the console port places you in the system execution space.

Note For ASAv console access, see the ASAv quick start guide.

Access the Appliance Console

Follow these steps to access the appliance console.

Procedure

Step 1 Connect a computer to the console port using the provided console cable, and connect to the console using a terminal emulator set for 9600 baud, 8 data bits, no parity, 1 stop bit, no flow control.

See the hardware guide for your ASA for more information about the console cable.

Step 2 Press the **Enter** key to see the following prompt:

```
ciscoasa>
```

This prompt indicates that you are in user EXEC mode. Only basic commands are available from user EXEC mode.

Step 3 Access privileged EXEC mode.

enable

You are prompted for the password. By default, the password is blank, and you can press the **Enter** key to continue. See Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503 to change the enable password.

Example:

```
ciscoasa> enable
Password:
ciscoasa#
```

All non-configuration commands are available in privileged EXEC mode. You can also enter configuration mode from privileged EXEC mode.

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 4 Access global configuration mode.

configure terminal

Example:

```
ciscoasa# configure terminal
ciscoasa(config)#
```

You can begin to configure the ASA from global configuration mode. To exit global configuration mode, enter the **exit**, **quit**, or **end** command.

Access the ASA Console on the Firepower 9300 Chassis

For initial configuration, access the command-line interface by connecting to the Firepower 9300 chassis supervisor (either to the console port or remotely using Telnet or SSH) and then connecting to the ASA security module.

Procedure

Step 1 Connect to the Firepower 9300 chassis supervisor CLI (console or SSH), and then session to the ASA:

connect module *slot* **console**

The first time you access the module, you access the FXOS module CLI. You must then connect to the ASA application.

connect asa

Example:

```
Firepower# connect module 1 console
Firepower-module1> connect asa

asa>
```

Step 2 Access privileged EXEC mode, which is the highest privilege level.

enable

You are prompted for the password. By default, the password is blank, and you can press the **Enter** key to continue. See Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503 to change the enable password.

Example:

```
asa> enable
Password:
asa#
```

All non-configuration commands are available in privileged EXEC mode. You can also enter configuration mode from privileged EXEC mode.

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 3 Enter global configuration mode.

configure terminal

Example:

```
asa# configure terminal
asa(config)#
```

To exit global configuration mode, enter the **disable**, **exit**, or **quit** command.

Step 4 Exit the application console to the FXOS module CLI by entering **Ctrl-a, d**

You might want to use the FXOS module CLI for troubleshooting purposes.

Step 5 Return to the supervisor level of the FXOS CLI.

a) Enter ~

   You exit to the Telnet application.

b) To exit the Telnet application, enter:

   telnet>**quit**

Access the ASA Services Module Console

For initial configuration, access the command-line interface by connecting to the switch (either to the console port or remotely using Telnet or SSH) and then connecting to the ASASM. This section describes how to access the ASASM CLI.

About Connection Methods

From the switch CLI, you can use two methods to connect to the ASASM:

- Virtual console connection—Using the **service-module session** command, you create a virtual console connection to the ASASM, with all the benefits and limitations of an actual console connection.

  Benefits include:

  - The connection is persistent across reloads and does not time out.
  - You can stay connected through ASASM reloads and view startup messages.
  - You can access ROMMON if the ASASM cannot load the image.
  - No initial password configuration is required.

  Limitations include:

  - The connection is slow (9600 baud).
  - You can only have one console connection active at a time.
  - You cannot use this command in conjunction with a terminal server where **Ctrl-Shift-6**, **x** is the escape sequence to return to the terminal server prompt. **Ctrl-Shift-6**, **x** is also the sequence to escape the ASASM console and return to the switch prompt. Therefore, if you try to exit the ASASM console in this situation, you instead exit all the way to the terminal server prompt. If you reconnect the terminal server to the switch, the ASASM console session is still active; you can never exit to the switch prompt. You must use a direct serial connection to return the console to the switch prompt. In this case, either change the terminal server or switch escape character in Cisco IOS software, or use the Telnet **session** command instead.

  Note Because of the persistence of the console connection, if you do not properly log out of the ASASM, the connection may exist longer than intended. If someone else wants to log in, they will need to kill the existing connection.

- Telnet connection—Using the **session** command, you create a Telnet connection to the ASASM.

  Note You cannot connect using this method for a new ASASM; this method requires you to configure a Telnet login password on the ASASM (there is no default password). After you set a password using the **passwd** command, you can use this method.

  Benefits include:

  - You can have multiple sessions to the ASASM at the same time.
  - The Telnet session is a fast connection.

  Limitations include:

  - The Telnet session is terminated when the ASASM reloads, and can time out.

- You cannot access the ASASM until it completely loads; you cannot access ROMMON.
- You must first set a Telnet login password; there is no default password.

Log Into the ASA Services Module

For initial configuration, access the command-line interface by connecting to the switch (either to the switch console port or remotely using Telnet or SSH) and then connecting to the ASASM.

If your system is already in multiple context mode, then accessing the ASASM from the switch places you in the system execution space.

Later, you can configure remote access directly to the ASASM using Telnet or SSH.

Procedure

Step 1 From the switch, perform one of the following:

- Available for initial access—From the switch CLI, enter this command to gain console access to the ASASM:

  service-module session [**switch** {**1** | **2**}] **slot** *number*

  Example:

  ```
  Router# service-module session slot 3
  ciscoasa>
  ```

  For a switch in a VSS, enter the **switch** argument.

  To view the module slot numbers, enter the **show module** command at the switch prompt.

  You access user EXEC mode.

- Available after you configure a login password—From the switch CLI, enter this command to Telnet to the ASASM over the backplane:

  session [**switch** {**1** | | **2**}] **slot** *number* **processor 1**

  You are prompted for the login password:

  ```
  ciscoasa passwd:
  ```

  Example:

  ```
  Router# session slot 3 processor 1
  ciscoasa passwd: cisco
  ciscoasa>
  ```

  For a switch in a VSS, enter the **switch** argument.

  The **session** *slot* **processor 0** command, which is supported on other services modules, is not supported on the ASASM; the ASASM does not have a processor 0.

  To view the module slot numbers, enter the **show module** command at the switch prompt.

Enter the login password to the ASASM. Set the password using the **passwd** command. There is no default password.

You access user EXEC mode.

Step 2 Access privileged EXEC mode, which is the highest privilege level.

enable

You are prompted for the password. By default, the password is blank, and you can press the **Enter** key to continue. See Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503 to change the enable password.

Example:

```
ciscoasa> enable
Password:
ciscoasa#
```

All non-configuration commands are available in privileged EXEC mode. You can also enter configuration mode from privileged EXEC mode.

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 3 Access global configuration mode:

configure terminal

To exit global configuration mode, enter the **disable**, **exit**, or **quit** command.

Related Topics

Guidelines for Management Access

Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503

Log Out of a Console Session

If you do not log out of the ASASM, the console connection persists; there is no timeout. To end the ASASM console session and access the switch CLI, perform the following steps.

To kill another user’s active connection, which may have been unintentionally left open, see Kill an Active Console Connection, on page 25.

Procedure

To return to the switch CLI, type the following:

Ctrl-Shift-6, **x**

You return to the switch prompt:

```
asasm# [Ctrl-Shift-6, x]
Router#
```

Note Shift-6 on US and UK keyboards issues the caret (^) character. If you have a different keyboard and cannot issue the caret (^) character as a standalone character, you can temporarily or permanently change the escape character to a different character. Use the **terminal escape-character** *ascii_number* command (to change for this session) or the **default escape-character** *ascii_number* command (to change permanently). For example, to change the sequence for the current session to **Ctrl-w**, **x**, enter **terminal escape-character 23**.

Kill an Active Console Connection

Because of the persistence of a console connection, if you do not properly log out of the ASASM, the connection may exist longer than intended. If someone else wants to log in, they will need to kill the existing connection.

Procedure

Step 1 From the switch CLI, show the connected users using the **show users** command. A console user is called "con". The Host address shown is 127.0.0.*slot*0, where *slot* is the slot number of the module.

show users

For example, the following command output shows a user "con" on line 0 on a module in slot 2:

```
Router# show users
Line        User       Host(s)              Idle       Location
*  0        con 0     127.0.0.20            00:00:02
```

Step 2 To clear the line with the console connection, enter the following command:

clear *line number*

For example:

```
Router# clear line 0
```

Log Out of a Telnet Session

To end the Telnet session and access the switch CLI, perform the following steps.

Procedure

To return to the switch CLI, type **exit** from the ASASM privileged or user EXEC mode. If you are in a configuration mode, enter **exit** repeatedly until you exit the Telnet session.

You return to the switch prompt:

```
asasm# exit
```

```
Router#
```

Note You can alternatively escape the Telnet session using the escape sequence **Ctrl-Shift-6**, **x**; this escape sequence lets you resume the Telnet session by pressing the **Enter** key at the switch prompt. To disconnect your Telnet session from the switch, enter **disconnect** at the switch CLI. If you do not disconnect the session, it will eventually time out according to the ASASM configuration.

Access the Software Module Console

If you have a software module installed, such as the ASA FirePOWER module on the ASA 5506-X, you can session to the module console.

Note You cannot access the *hardware* module CLI over the ASA backplane using the **session** command.

Procedure

From the ASA CLI, session to the module:

session {**sfr** | **cxsc** | **ips**} **console**

Example:

```
ciscoasa# session sfr console
Opening console session with module sfr.
Connected to module sfr. Escape character sequence is 'CTRL-^X'.

Cisco ASA SFR Boot Image 5.3.1
asasfr login: admin
Password: Admin123
```

Access the ASA 5506W-X Wireless Access Point Console

To access the wireless access point console, perform the following steps.

Procedure

Step 1 From the ASA CLI, session to the access point:

session wlan console

Example:

```
ciscoasa# session wlan console
```

```
opening console session with module wlan
connected to module wlan. Escape character sequence is 'CTRL-^X'

ap>
```

Step 2 See the Cisco IOS Configuration Guide for Autonomous Aironet Access Points for information about the access point CLI.

Configure ASDM Access

This section describes how to access ASDM with a default configuration and how to configure access if you do not have a default configuration.

Use the Factory Default Configuration for ASDM Access (Appliances, ASAv)

With a factory default configuration, ASDM connectivity is pre-configured with default network settings.

Procedure

Connect to ASDM using the following interface and network settings:

- The management interface depends on your model:
    - Firepower 9300—The Management type interface and IP address of your choice defined when you deployed. Management hosts are allowed from any network.
    - ASA 5506-X, ASA 5508-X, and ASA 5516-X—Inside GigabitEthernet 1/2 (192.168.1.1) and for ASA 5506W-X, wifi GigabitEthernet 1/9 (192.168.10.1). Inside hosts are limited to the 192.168.1.0/24 network, and wifi hosts are limited to 192.168.10.0/24.
    - ASA 5512-X and higher—Management 0/0 (192.168.1.1). Management hosts are limited to the 192.168.1.0/24 network.
    - ASAv—Management 0/0 (set during deployment). Management hosts are limited to the management network.
    - ISA 3000—Management 1/1 (192.168.1.1). Management hosts are limited to the 192.168.1.0/24 network.

Note If you change to multiple context mode, you can access ASDM from the admin context using the network settings above.

Related Topics

Factory Default Configurations, on page 34

Enable or Disable Multiple Context Mode, on page 185

Start ASDM, on page 32

Customize ASDM Access

This procedure applies to all models except the ASA Services Module.

Use this procedure if *one or more* of the following conditions applies:

- You do not have a factory default configuration
- You want to change the management IP address
- You want to change to transparent firewall mode
- You want to change to multiple context mode

For routed, single mode, for quick and easy ASDM access, we recommend applying the factory default configuration with the option to set your own management IP address. Use the procedure in this section only if you have special needs such as setting transparent or multiple context mode, or if you have other configuration that you need to preserve.

Note For the ASAv, you can configure transparent mode when you deploy, so this procedure is primarily useful after you deploy if you need to clear your configuration, for example.

Procedure

Step 1 Access the CLI at the console port.

Step 2 (Optional) Enable transparent firewall mode:

This command clears your configuration.

firewall transparent

Step 3 Configure the management interface:

```
interface interface_id
   nameif name
   security-level level
   no shutdown
   ip address ip_address mask
```

Example:

```
ciscoasa(config)# interface management 0/0
ciscoasa(config-if)# nameif management
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# no shutdown
ciscoasa(config-if)# ip address 192.168.1.1 255.255.255.0
```

The **security-level** is a number between 1 and 100, where 100 is the most secure.

Step 4 (For directly-connected management hosts) Set the DHCP pool for the management network:

```
dhcpd address ip_address-ip_address interface_name
```

dhcpd enable *interface_name*

Example:

```
ciscoasa(config)# dhcpd address 192.168.1.2-192.168.1.254 management
ciscoasa(config)# dhcpd enable management
```

Make sure you do not include the interface address in the range.

Step 5 (For remote management hosts) Configure a route to the management hosts:

route *management_ifc management_host_ip mask gateway_ip* **1**

Example:

```
ciscoasa(config)# route management 10.1.1.0 255.255.255.0 192.168.1.50 1
```

Step 6 Enable the HTTP server for ASDM:

http server enable

Step 7 Allow the management host(s) to access ASDM:

http *ip_address mask interface_name*

Example:

```
ciscoasa(config)# http 192.168.1.0 255.255.255.0 management
```

Step 8 Save the configuration:

write memory

Step 9 (Optional) Set the mode to multiple mode:

mode multiple

When prompted, confirm that you want to convert the existing configuration to be the admin context. You are then prompted to reload the ASA.

Examples

The following configuration converts the firewall mode to transparent mode, configures the Management 0/0 interface, and enables ASDM for a management host:

```
firewall transparent
interface management 0/0

ip address 192.168.1.1 255.255.255.0
nameif management
security-level 100
no shutdown

dhcpd address 192.168.1.2-192.168.1.254 management
dhcpd enable management
http server enable
http 192.168.1.0 255.255.255.0 management
```

Related Topics

Restore the Factory Default Configuration, on page 35

Set the Firewall Mode, on page 157

Access the Appliance Console, on page 19

Start ASDM, on page 32

Configure ASDM Access for the ASA Services Module

Because the ASASM does not have physical interfaces, it does not come pre-configured for ASDM access; you must configure ASDM access using the CLI on the ASASM. To configure the ASASM for ASDM access, perform the following steps.

Before you begin

Assign a VLAN interface to the ASASM according to ASASM quick start guide.

Procedure

Step 1 Connect to the ASASM and access global configuration mode.

Step 2 (Optional) Enable transparent firewall mode:

firewall transparent

This command clears your configuration.

Step 3 Do one of the following to configure a management interface, depending on your mode:

- Routed mode—Configure an interface in routed mode:

```
interface vlan number
   ip address ip_address [mask]
   nameif name
   security-level level
```

  Example:

```
ciscoasa(config)# interface vlan 1
ciscoasa(config-if)# ip address 192.168.1.1 255.255.255.0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

  The **security-level** is a number between 1 and 100, where 100 is the most secure.

- Transparent mode—Configure a bridge virtual interface and assigns a management VLAN to the bridge group:

```
interface bvi number
```

```
   ip address ip_address [mask]

interface vlan number
   bridge-group bvi_number
   nameif name
   security-level level
```

Example:

```
ciscoasa(config)# interface bvi 1
ciscoasa(config-if)# ip address 192.168.1.1 255.255.255.0

ciscoasa(config)# interface vlan 1
ciscoasa(config-if)# bridge-group 1
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

The **security-level** is a number between 1 and 100, where 100 is the most secure.

Step 4 (For directly-connected management hosts) Enable DHCP for the management host on the management interface network:

```
dhcpd address ip_address-ip_address interface_name
dhcpd enable interface_name
```

Example:

```
ciscoasa(config)# dhcpd address 192.168.1.2-192.168.1.254 inside
ciscoasa(config)# dhcpd enable inside
```

Make sure you do not include the management address in the range.

Step 5 (For remote management hosts) Configure a route to the management hosts:

route *management_ifc management_host_ip mask gateway_ip* **1**

Example:

```
ciscoasa(config)# route management 10.1.1.0 255.255.255.0 192.168.1.50
```

Step 6 Enable the HTTP server for ASDM:

http server enable

Step 7 Allow the management host to access ASDM:

http *ip_address mask interface_name*

Example:

```
ciscoasa(config)# http 192.168.1.0 255.255.255.0 management
```

Step 8 Save the configuration:

write memory

Step 9 (Optional) Set the mode to multiple mode:

mode multiple

When prompted, confirm that you want to convert the existing configuration to be the admin context. You are then prompted to reload the ASASM.

Examples

The following routed mode configuration configures the VLAN 1 interface and enables ASDM for a management host:

```
interface vlan 1
nameif inside
ip address 192.168.1.1 255.255.255.0
security-level 100

dhcpd address 192.168.1.3-192.168.1.254 inside
dhcpd enable inside
http server enable
http 192.168.1.0 255.255.255.0 inside
```

The following configuration converts the firewall mode to transparent mode, configures the VLAN 1 interface and assigns it to BVI 1, and enables ASDM for a management host:

```
firewall transparent
interface bvi 1

ip address 192.168.1.1 255.255.255.0
interface vlan 1
bridge-group 1
nameif inside
security-level 100

dhcpd address 192.168.1.3-192.168.1.254 inside
dhcpd enable inside
http server enable
http 192.168.1.0 255.255.255.0 inside
```

Related Topics

Access the ASA Services Module Console, on page 21
About Connection Methods, on page 22
Log Out of a Console Session, on page 24
Kill an Active Console Connection, on page 25
Log Out of a Telnet Session, on page 25
Set the Firewall Mode, on page 157

Start ASDM

You can start ASDM using two methods:

- ASDM-IDM Launcher—The Launcher is an application downloaded from the ASA using a web browser that you can use to connect to any ASA IP address. You do not need to re-download the launcher if you want to connect to other ASAs.
- Java Web Start—For each ASA that you manage, you need to connect with a web browser and then save or launch the Java Web Start application. You can optionally save the shortcut to your computer; however you need separate shortcuts for each ASA IP address.

Note If you use web start, clear the Java cache or you might lose changes to some pre-login policies such as Hostscan. This problem does not occur if you use the launcher.

Within ASDM, you can choose a different ASA IP address to manage; the difference between the Launcher and Java Web Start functionality rests primarily in how you initially connect to the ASA and launch ASDM.

This section describes how to connect to ASDM initially, and then launch ASDM using the Launcher or the Java Web Start.

ASDM stores files in the local \Users\<user_id>\.asdm directory, including cache, log, and preferences, and also in the Temp directory, including AnyConnect profiles.

Procedure

Step 1 On the computer that you specified as the ASDM client, enter the following URL:

https://*asa_ip_address***/admin**

Note Be sure to specify **https://**, and not **http://** or just the IP address (which defaults to HTTP); the ASA does not automatically forward an HTTP request to HTTPS.

The ASDM launch page appears with the following buttons:

- **Install ASDM Launcher and Run ASDM**
- **Run ASDM**
- **Run Startup Wizard**

Step 2 To download the Launcher:

a) Click **Install ASDM Launcher and Run ASDM**.
b) Leave the username and password fields empty (for a new installation), and click **OK**. With no HTTPS authentication configured, you can gain access to ASDM with no username and the **enable** password, which is blank by default. **Note**: If you enabled HTTPS authentication, enter your username and associated password. Even without authentication, if you enter a username and password at the login screen (instead of leaving the username blank), ASDM checks the local database for a match.
c) Save the installer to your computer, and then start the installer. The ASDM-IDM Launcher opens automatically after installation is complete.
d) Enter the management IP address, the same username and password (blank for a new installation), and then click **OK**.

Step 3 To use Java Web Start:

a) Click **Run ASDM** or **Run Startup Wizard**.

b) Save the shortcut to your computer when prompted. You can optionally open it instead of saving it.
c) Start Java Web Start from the shortcut.
d) Accept any certificates according to the dialog boxes that appear. The Cisco ASDM-IDM Launcher appears.
e) Leave the username and password fields empty (for a new installation), and click **OK**. With no HTTPS authentication configured, you can gain access to ASDM with no username and the **enable** password, which is blank by default. **Note**: If you enabled HTTPS authentication, enter your username and associated password. Even without authentication, if you enter a username and password at the login screen (instead of leaving the username blank), ASDM checks the local database for a match.

Factory Default Configurations

The factory default configuration is the configuration applied by Cisco to new ASAs.

- ASA 5506-X, 5508-X and 5516-X—The factory default configuration enables a functional inside/outside configuration. You can manage the ASA using ASDM from the inside interface.
- ASA 5512-X through ASA 5585-X—The factory default configuration configures an interface for management so that you can connect to it using ASDM, with which you can then complete your configuration.
- Firepower 9300 chassis—When you deploy the standalone or cluster of ASAs, the factory default configuration configures an interface for management so that you can connect to it using ASDM, with which you can then complete your configuration.
- ASAv—Depending on your hypervisor, as part of deployment, the deployment configuration (the initial virtual deployment settings) configures an interface for management so that you can connect to it using ASDM, with which you can then complete your configuration. You can also configure failover IP addresses. You can also apply a “factory default” configuration if desired.
- ASASM—No default configuration. See Access the ASA Services Module Console, on page 21 to start configuration.
- ISA 3000—The factory default configuration is an almost-complete transparent firewall mode configuration with all inside and outside interfaces on the same network; you can connect to the management interface with ASDM to set the IP address of your network. Hardware bypass is enabled for two interface pairs, and all traffic is sent to the ASA FirePOWER module in Inline Tap Monitor-Only Mode. This mode sends a duplicate stream of traffic to the ASA Firepower module for monitoring purposes only.

For appliances and the Firepower 9300 chassis, the factory default configuration is available only for routed firewall mode and single context mode, except for the ISA 3000, where the factory default configuration is only available in transparent mode. For the ASAv, you can choose transparent or routed mode at deployment.

Note In addition to the image files and the (hidden) default configuration, the following folders and files are standard in flash memory: log/, crypto_archive/, and coredumpinfo/coredump.cfg. The date on these files may not match the date of the image files in flash memory. These files aid in potential troubleshooting; they do not indicate that a failure has occurred.

Restore the Factory Default Configuration

This section describes how to restore the factory default configuration. For the ASAv, this procedure erases the deployment configuration and applies the same factory default configuration as for the ASA 5525-X.

Note On the ASASM, restoring the factory default configuration simply erases the configuration; there is no factory default configuration.

On the Firepower 9300, restoring the factory default configuration simply erases the configuration; to restore the default configuration, you must re-deploy the ASA from the supervisor.

Before you begin

This feature is available only in routed firewall mode, except for the ISA 3000, where this command is only supported in transparent mode. In addition, this feature is available only in single context mode; an ASA with a cleared configuration does not have any defined contexts to configure automatically using this feature.

Procedure

Step 1 Restore the factory default configuration:

configure factory-default [*ip_address* [*mask*]]

Example:

```
ciscoasa(config)# configure factory-default 10.1.1.1 255.255.255.0
```

If you specify the *ip_address*, then you set the inside or management interface IP address, depending on your model, instead of using the default IP address. See the following model guidelines for which interface is set by the *ip_address* option:

- Firepower 9300—No effect.
- ASAv—Sets the **management** interface IP address.
- ASA 5506-X—Sets the **inside** interface IP address.
- ASA 5508-X and 5516-X—Sets the **inside** interface IP address.
- ASA 5512-X, 5515-X, 5525-X, 5545-X, 5555-X—Sets the **management** interface IP address.
- ASA 5585-X—Sets the **management** interface IP address.
- ISA 3000—Sets the **management** interface IP address.
- ASASM—No effect.

The **http** command uses the subnet you specify. Similarly, the **dhcpd address** command range consists of all available addresses higher than the IP address you specify. For example, if you specify 10.5.6.78 with a subnet mask of 255.255.255.0, then the DHCP address range will be 10.5.6.79-10.5.6.254.

For the Firepower 2100: This model does not use the **boot system** command; packages are managed by FXOS.

For all other models: This command clears the **boot system** command, if present, along with the rest of the configuration. The **boot system** command lets you boot from a specific image. The next time you reload the ASA after restoring the factory configuration, it boots from the first image in internal flash memory; if you do not have an image in internal flash memory, the ASA does not boot.

Example:

```
docs-bxb-asa3(config)# configure factory-default 10.86.203.151 255.255.254.0
Based on the management IP address and mask, the DHCP address
pool size is reduced to 103 from the platform limit 256

WARNING: The boot system configuration will be cleared.
The first image found in disk0:/ will be used to boot the
system on the next reload.
Verify there is a valid image on disk0:/ or the system will
not boot.

Begin to apply factory-default configuration:
Clear all configuration
WARNING: The new maximum-session limit will take effect after the running-config is saved
and the system boots next time. Command accepted
WARNING: Local user database is empty and there are still 'aaa' commands for 'LOCAL'.
Executing command: interface management0/0
Executing command: nameif management
INFO: Security level for "management" set to 0 by default.
Executing command: ip address 10.86.203.151 255.255.254.0
Executing command: security-level 100
Executing command: no shutdown
Executing command: exit
Executing command: http server enable
Executing command: http 10.86.202.0 255.255.254.0 management
Executing command: dhcpd address 10.86.203.152-10.86.203.254 management
Executing command: dhcpd enable management
Executing command: logging asdm informational
Factory-default configuration is completed
ciscoasa(config)#
```

Step 2 Save the default configuration to flash memory:

write memory

This command saves the running configuration to the default location for the startup configuration, even if you previously configured the **boot** config command to set a different location; when the configuration was cleared, this path was also cleared.

Restore the ASAv Deployment Configuration

This section describes how to restore the ASAv deployment (Day 0) configuration.

Procedure

Step 1 For failover, power off the standby unit.

To prevent the standby unit from becoming active, you must power it off. If you leave it on, when you erase the active unit configuration, then the standby unit becomes active. When the former active unit reloads and reconnects over the failover link, the old configuration will sync from the new active unit, wiping out the deployment configuration you wanted.

Step 2 Restore the deployment configuration after you reload. For failover, enter this command on the active unit:

write erase

Note The ASAv boots the current running image, so you are not reverted to the original boot image. To use the original boot image, see the **boot image** command.

Do not save the configuration.

Step 3 Reload the ASAv and load the deployment configuration:

reload

Step 4 For failover, power on the standby unit.

After the active unit reloads, power on the standby unit. The deployment configuration will sync to the standby unit.

ASA 5506-X, 5508-X, and 5516-X Default Configuration

The default factory configuration for the ASA 5506-X series, 5508-X, and 5516-X configures the following:

- **inside --> outside** traffic flow—GigabitEthernet 1/1 (outside), GigabitEthernet 1/2 (inside)
- **outside IP address** from DHCP
- **inside IP address**—192.168.1.1
- (ASA 5506W-X) **wifi <--> inside, wifi --> outside** traffic flow—GigabitEthernet 1/9 (wifi)
- (ASA 5506W-X) **wifi IP address**—192.168.10.1
- **DHCP server** on inside and wifi. The access point itself and all its clients use the ASA as the DHCP server.
- **Default route** from outside DHCP
- **Management 1/1 interface** is Up, but otherwise unconfigured. The ASA FirePOWER module can then use this interface to access the ASA inside network and use the inside interface as the gateway to the Internet.
- **ASDM** access—inside and wifi hosts allowed.
- **NAT**—Interface PAT for all traffic from inside, wifi, and management to outside.

The configuration consists of the following commands:

```
interface Management1/1
  management-only
  no nameif
  no security-level
  no ip address
  no shutdown
```

```
interface GigabitEthernet1/1
  nameif outside
  security-level 0
  ip address dhcp setroute
  no shutdown
interface GigabitEthernet1/2
  nameif inside
  security-level 100
  ip address 192.168.1.1 255.255.255.0
  no shutdown
!
object network obj_any
  subnet 0.0.0.0 0.0.0.0
  nat (any,outside) dynamic interface
!
http server enable
http 192.168.1.0 255.255.255.0 inside
!
dhcpd auto_config outside
dhcpd address 192.168.1.5-192.168.1.254 inside
dhcpd enable inside
!
logging asdm informational
```

For the ASA 5506W-X, the following commands are also included:

```
same-security-traffic permit inter-interface
!
interface GigabitEthernet 1/9
  security-level 100
  nameif wifi
  ip address 192.168.10.1 255.255.255.0
  no shutdown
!
http 192.168.10.0 255.255.255.0 wifi
!
dhcpd address 192.168.10.2-192.168.10.254 wifi
dhcpd enable wifi
```

ASA 5512-X through ASA 5585-X Default Configuration

The default factory configuration for the ASA 5512-X through ASA 5585-X configures the following:

- Management interface—Management 0/0 (management).
- IP address—The management address is 192.168.1.1/24.
- DHCP server—Enabled for management hosts so that a computer connecting to the management interface receives an address between 192.168.1.2 and 192.168.1.254.
- ASDM access—Management hosts allowed.

The configuration consists of the following commands:

```
interface management 0/0
  ip address 192.168.1.1 255.255.255.0
  nameif management
  security-level 100
  no shutdown
```

```
!
asdm logging informational
asdm history enable
!
http server enable
http 192.168.1.0 255.255.255.0 management
!
dhcpd address 192.168.1.2-192.168.1.254 management
dhcpd enable management
```

Firepower 9300 Chassis Default Configuration

When you deploy the ASA on the Firepower 9300 chassis, you can pre-set many parameters that let you connect to the Management interface using ASDM. A typical configuration includes the following settings:

- Management interface:
  - Management type interface of your choice defined on the Firepower 9300 Chassis supervisor
  - Named “management”
  - IP address of your choice
  - Security level 0
  - Management-only
- Default route through the management interface
- ASDM access—All hosts allowed.

The configuration for a standalone unit consists of the following commands. For additional configuration for clustered units, see Create an ASA Cluster, on page 362.

```
interface <management_ifc>
  management-only
  ip address <ip_address> <mask>
  ipv6 address <ipv6_address>
  ipv6 enable
  nameif management
  security-level 0
  no shutdown
!
http server enable
http 0.0.0.0 0.0.0.0 management
http ::/0 management
!
route management 0.0.0.0 0.0.0.0 <gateway_ip> 1
ipv6 route management ::/0 <gateway_ipv6>
```

ISA 3000 Default Configuration

The default factory configuration for the ISA 3000 configures the following:

- **Transparent firewall mode**—A transparent firewall is a Layer 2 firewall that acts like a “bump in the wire,” or a “stealth firewall,” and is not seen as a router hop to connected devices.

- **1 Bridge Virtual Interface**—All member interfaces are in the same network (**IP address *not* pre-configured; you must set to match your network**): GigabitEthernet 1/1 (outside1), GigabitEthernet 1/2 (inside1), GigabitEthernet 1/3 (outside2), GigabitEthernet 1/4 (inside2)
- All **inside and outside** interfaces can communicate with each other.
- **Management 1/1** interface—192.168.1.1/24 for ASDM access.
- **DHCP** for clients on management.
- **ASDM** access—Management hosts allowed.
- **Hardware bypass** is enabled for the following interface pairs: GigabitEthernet 1/1 & 1/2; GigabitEthernet 1/3 & 1/4

Note When the ISA 3000 loses power and goes into hardware bypass mode, only the above interface pairs can communicate; inside1 and inside2, and outside1 and outside2 can no longer communicate. Any existing connections between these interfaces will be lost. When the power comes back on, there is a brief connection interruption as the ASA takes over the flows.

- **ASA FirePOWER module**—All traffic is sent to the module in Inline Tap Monitor-Only Mode. This mode sends a duplicate stream of traffic to the ASA Firepower module for monitoring purposes only.

The configuration consists of the following commands:

```
firewall transparent

interface GigabitEthernet1/1
  bridge-group 1
  nameif outside1
  security-level 0
  no shutdown
interface GigabitEthernet1/2
  bridge-group 1
  nameif inside1
  security-level 100
  no shutdown
interface GigabitEthernet1/3
  bridge-group 1
  nameif outside2
  security-level 0
  no shutdown
interface GigabitEthernet1/4
  bridge-group 1
  nameif inside2
  security-level 100
  no shutdown
interface Management1/1
  management-only
  no shutdown
  nameif management
  security-level 100
  ip address 192.168.1.1 255.255.255.0
interface BVI1
  no ip address
```

```
access-list allowAll extended permit ip any any
access-group allowAll in interface outside1
access-group allowAll in interface outside2

same-security-traffic permit inter-interface

hardware-bypass GigabitEthernet 1/1-1/2
hardware-bypass GigabitEthernet 1/3-1/4

http server enable
http 192.168.1.0 255.255.255.0 management

dhcpd address 192.168.1.5-192.168.1.254 management
dhcpd enable management


access-list sfrAccessList extended permit ip any any
class-map sfrclass
  match access-list sfrAccessList
policy-map global_policy
  class sfrclass
  sfr fail-open monitor-only
service-policy global_policy global
```

ASAv Deployment Configuration

When you deploy the ASAv, you can pre-set many parameters that let you connect to the Management 0/0 interface using ASDM. A typical configuration includes the following settings:

- Routed or Transparent firewall mode
- Management 0/0 interface:
  - Named “management”
  - IP address or DHCP
  - Security level 0
  - Management-only
- Static route for the management host IP address (if it is not on the management subnet)
- HTTP server enabled or disabled
- HTTP access for the management host IP address
- (Optional) Failover link IP addresses for GigabitEthernet 0/8, and the Management 0/0 standby IP address
- DNS server
- Smart licensing ID token
- Smart licensing Throughput Level and Standard Feature Tier
- (Optional) Smart Call Home HTTP Proxy URL and port
- (Optional) SSH management settings:
  - Client IP addresses

  - Local username and password
  - Authentication required for SSH using the LOCAL database

- (Optional) REST API enabled or disabled

Note To successfully register the ASAv with the Cisco Licensing Authority, the ASAv requires Internet access. You might need to perform additional configuration after deployment to achieve Internet access and successful license registration.

See the following sample configuration for a standalone unit:

```
interface Management0/0
  nameif management
  security-level 0
  ip address ip_address
  management-only
  no shutdown
http server enable
http managemment_host_IP mask management
route management management_host_IP mask gateway_ip 1
dns server-group DefaultDNS
  name-server ip_address
call-home
  http-proxy ip_address port port
license smart
  feature tier standard
  throughput level {100M | 1G | 2G}
  license smart register idtoken id_token
aaa authentication ssh console LOCAL
username username password password
ssh source_IP_address mask management
rest-api image boot:/path
rest-api agent
```

See the following sample configuration for a primary unit in a failover pair:

```
nameif management
  security-level 0
  ip address ip_address standby standby_ip
  management-only
  no shutdown
route management management_host_IP mask gateway_ip 1
http server enable
http managemment_host_IP mask management
dns server-group DefaultDNS
  name-server ip_address
call-home
  http-proxy ip_address port port
license smart
  feature tier standard
  throughput level {100M | 1G | 2G}
  license smart register idtoken id_token
aaa authentication ssh console LOCAL
username username password password
```

```
ssh source_IP_address mask management
rest-api image boot:/path
rest-api agent
failover
failover lan unit primary
failover lan interface fover gigabitethernet0/8
failover link fover gigabitethernet0/8
failover interface ip fover primary_ip mask standby standby_ip
```

Work with the Configuration

This section describes how to work with the configuration. The ASA loads the configuration from a text file, called the startup configuration. This file resides by default as a hidden file in internal flash memory. You can, however, specify a different path for the startup configuration.

When you enter a command, the change is made only to the running configuration in memory. You must manually save the running configuration to the startup configuration for your changes to remain after a reboot.

The information in this section applies to both single and multiple security contexts, except where noted.

Save Configuration Changes

This section describes how to save your configuration.

Save Configuration Changes in Single Context Mode

To save the running configuration to the startup configuration, perform the following procedure.

Procedure

Save the running configuration to the startup configuration:

write memory

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

Save Configuration Changes in Multiple Context Mode

You can save each context (and system) configuration separately, or you can save all context configurations at the same time.

Save Each Context and System Separately

Use the following procedure to save the system or context configuration.

Procedure

From within the context or the system, save the running configuration to the startup configuration:

write memory

For multiple context mode, context startup configurations can reside on external servers. In this case, the ASA saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

Save All Context Configurations at the Same Time

Use the following procedure to save all context configurations at the same time, as well as the system configuration.

Procedure

From the system execution space, save the running configuration to the startup configuration for all contexts and the system configuration:

write memory all [**/noconfirm**]

If you do not enter the **/noconfirm** keyword, you see the following prompt:

```
Are you sure [Y/N]:
```

After you enter **Y**, the ASA saves the system configuration and each context. Context startup configurations can reside on external servers. In this case, the ASA saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

After the ASA saves each context, the following message appears:

```
‘Saving context ‘b’ ... ( 1/3 contexts saved ) ’
```

Sometimes, a context is not saved because of an error. See the following information for errors:

- For contexts that are not saved because of low memory, the following message appears:

  ```
  The context 'context a' could not be saved due to Unavailability of resources
  ```

- For contexts that are not saved because the remote destination is unreachable, the following message appears:

  ```
  The context 'context a' could not be saved due to non-reachability of destination
  ```

- For contexts that are not saved because the context is locked, the following message appears:

  ```
  Unable to save the configuration for the following contexts as these contexts are locked.
  context ‘a’ , context ‘x’ , context ‘z’ .
  ```

A context is only locked if another user is already saving the configuration or in the process of deleting the context.

- For contexts that are not saved because the startup configuration is read-only (for example, on an HTTP server), the following message report is printed at the end of all other messages:

```
Unable to save the configuration for the following contexts as these contexts have
read-only config-urls:
context ‘a’ , context ‘b’ , context ‘c’ .
```

- For contexts that are not saved because of bad sectors in the flash memory, the following message appears:

```
The context 'context a' could not be saved due to Unknown errors
```

Copy the Startup Configuration to the Running Configuration

Use one of the following commands to copy a new startup configuration to the running configuration:

- **copy startup-config running-config**

  Merges the startup configuration with the running configuration. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

- **reload**

  Reloads the ASA, which loads the startup configuration and discards the running configuration.

- **clear configure all** and then **copy startup-config running-config**

  Loads the startup configuration and discards the running configuration without requiring a reload.

View the Configuration

The following commands let you view the running and startup configurations:

- **show running-config**

  Views the running configuration.

- **show running-config** *command*

  Views the running configuration of a specific command.

- **show startup-config**

  Views the startup configuration.

Clear and Remove Configuration Settings

To erase settings, enter one of the following commands:

- **clear configure** *configurationcommand* [*level2configurationcommand*]

  Clears all the configuration for a specified command. If you only want to clear the configuration for a specific version of the command, you can enter a value for *level2configurationcommand*.

  For example, to clear the configuration for all **aaa** commands, enter the following command:

  ```
  ciscoasa(config)# clear configure aaa
  ```

  To clear the configuration for only **aaa authentication** commands, enter the following command:

  ```
  ciscoasa(config)# clear configure aaa authentication
  ```

- **no** *configurationcommand* [*level2configurationcommand*] *qualifier*

  Disables the specific parameters or options of a command. In this case, you use the **no** command to remove the specific configuration identified by *qualifier*.

  For example, to remove a specific **access-list** command, enter enough of the command to identify it uniquely; you may have to enter the entire command:

  ```
  ciscoasa(config)# no access-list abc extended permit icmp any any object-group obj_icmp_1
  ```

- **write erase**

  Erases the startup configuration.

  Note For the ASAv, this command restores the deployment configuration after a reload. To erase the configuration completely, use the **clear configure all** command.

- **clear configure all**

  Erases the running configuration.

  Note In multiple context mode, if you enter **clear configure all** from the system configuration, you also remove all contexts and stop them from running. The context configuration files are not erased, and remain in their original location.

Note

For the Firepower 2100: This model does not use the **boot system** command; packages are managed by FXOS.

For all other models: This command clears the **boot system** command, if present, along with the rest of the configuration. The **boot** system command lets you boot from a specific image, including an image on the external flash memory card. The next time you reload the ASA, it boots from the first image in internal flash memory; if you do not have an image in internal flash memory, the ASA does not boot.

Create Text Configuration Files Offline

This guide describes how to use the CLI to configure the ASA; when you save commands, the changes are written to a text file. Instead of using the CLI, however, you can edit a text file directly on your computer and paste a configuration at the configuration mode command-line prompt in its entirety, or line by line. Alternatively, you can download a text file to the ASA internal flash memory. See Software and Configurations, on page 875 for information on downloading the configuration file to the ASA.

In most cases, commands described in this guide are preceded by a CLI prompt. The prompt in the following example is “ciscoasa(config)#”:

```
ciscoasa(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted as follows:

```
context a
```

For additional information about formatting the file, see Using the Command-Line Interface, on page 1055.

Apply Configuration Changes to Connections

When you make security policy changes to the configuration, all *new* connections use the new security policy. Existing connections continue to use the policy that was configured at the time of the connection establishment. **show** command output for old connections reflect the old configuration, and in some cases will not include data about the old connections.

For example, if you remove a QoS **service-policy** from an interface, then re-add a modified version, then the **show service-policy** command only displays QoS counters associated with new connections that match the new service policy; existing connections on the old policy no longer show in the command output.

To ensure that all connections use the new policy, you need to disconnect the current connections so that they can reconnect using the new policy.

To disconnect connections, enter one of the following commands:

- **clear local-host** [*ip_address*] [**all**]

This command reinitializes per-client run-time states such as connection limits and embryonic limits. As a result, this command removes any connection that uses those limits. See the **show local-host all** command to view all current connections per host.

With no arguments, this command clears all affected through-the-box connections. To also clear to-the-box connections (including your current management session), use the **all** keyword. To clear connections to and from a particular IP address, use the *ip_address* argument.

- **clear conn** [**all**] [**protocol** {**tcp** | **udp**}] [**address** *src_ip* [-*src_ip*] [**netmask** *mask*]] [**port** *src_port* [-*src_port*]] [**address** *dest_ip* [-*dest_ip*] [**netmask** *mask*]] [**port** *dest_port* [-*dest_port*]]

  This command terminates connections in any state. See the **show conn** command to view all current connections.

  With no arguments, this command clears all through-the-box connections. To also clear to-the-box connections (including your current management session), use the **all** keyword. To clear specific connections based on the source IP address, destination IP address, port, and/or protocol, you can specify the desired options.

Reload the ASA

To reload the ASA, complete the following procedure.

Procedure

Reload the ASA:

reload

Note In multiple context mode, you can only reload from the system execution space.

CHAPTER 3

Licenses: Product Authorization Key Licensing

A license specifies the options that are enabled on a given Cisco ASA. This document describes product authorization key (PAK) licenses for all physical ASAs. For the ASAv, see Licenses: Smart Software Licensing (ASAv, ASA on Firepower), on page 109.

About PAK Licenses

A license specifies the options that are enabled on a given ASA. It is represented by an activation key that is a 160-bit (5 32-bit words or 20 bytes) value. This value encodes the serial number (an 11 character string) and the enabled features.

Preinstalled License

By default, your ASA ships with a license already installed. This license might be the Base License, to which you want to add more licenses, or it might already have all of your licenses installed, depending on what you ordered and what your vendor installed for you.

Related Topics

Monitoring PAK Licenses, on page 89

Permanent License

You can have one permanent activation key installed. The permanent activation key includes all licensed features in a single key. If you also install time-based licenses, the ASA combines the permanent and time-based licenses into a running license.

Related Topics

How Permanent and Time-Based Licenses Combine, on page 50

Time-Based Licenses

In addition to permanent licenses, you can purchase time-based licenses or receive an evaluation license that has a time-limit. For example, you might buy a time-based AnyConnect Premium license to handle short-term surges in the number of concurrent SSL VPN users, or you might order a Botnet Traffic Filter time-based license that is valid for 1 year.

Note The ASA 5506-X and ASA 5506W-X do not support time-based licenses.

Time-Based License Activation Guidelines

- You can install multiple time-based licenses, including multiple licenses for the same feature. However, only one time-based license per feature can be *active* at a time. The inactive license remains installed, and ready for use. For example, if you install a 1000-session AnyConnect Premium license, and a 2500-session AnyConnect Premium license, then only one of these licenses can be active.
- If you activate an evaluation license that has multiple features in the key, then you cannot also activate another time-based license for one of the included features. For example, if an evaluation license includes the Botnet Traffic Filter and a 1000-session AnyConnect Premium license, you cannot also activate a standalone time-based 2500-session AnyConnect Premium license.

How the Time-Based License Timer Works

- The timer for the time-based license starts counting down when you activate it on the ASA.
- If you stop using the time-based license before it times out, then the timer halts. The timer only starts again when you reactivate the time-based license.
- If the time-based license is active, and you shut down the ASA, then the timer stops counting down. The time-based license only counts down when the ASA is running. The system clock setting does not affect the license; only ASA uptime counts towards the license duration.

How Permanent and Time-Based Licenses Combine

When you activate a time-based license, then features from both permanent and time-based licenses combine to form the running license. How the permanent and time-based licenses combine depends on the type of license. The following table lists the combination rules for each feature license.

Note Even when the permanent license is used, if the time-based license is active, it continues to count down.

Table 1: Time-Based License Combination Rules

Time-Based Feature	Combined License Rule
AnyConnect Premium Sessions	The higher value is used, either time-based or permanent. For example, if the permanent license is 1000 sessions, and the time-based license is 2500 sessions, then 2500 sessions are enabled. Typically, you will not install a time-based license that has less capability than the permanent license, but if you do so, then the permanent license is used.
Unified Communications Proxy Sessions	The time-based license sessions are added to the permanent sessions, up to the platform limit. For example, if the permanent license is 2500 sessions, and the time-based license is 1000 sessions, then 3500 sessions are enabled for as long as the time-based license is active.
Security Contexts	The time-based license contexts are added to the permanent contexts, up to the platform limit. For example, if the permanent license is 10 contexts, and the time-based license is 20 contexts, then 30 contexts are enabled for as long as the time-based license is active.
Botnet Traffic Filter	There is no permanent Botnet Traffic Filter license available; the time-based license is used.
All Others	The higher value is used, either time-based or permanent. For licenses that have a status of enabled or disabled, then the license with the enabled status is used. For licenses with numerical tiers, the higher value is used. Typically, you will not install a time-based license that has less capability than the permanent license, but if you do so, then the permanent license is used.

Related Topics

Monitoring PAK Licenses, on page 89

Stacking Time-Based Licenses

In many cases, you might need to renew your time-based license and have a seamless transition from the old license to the new one. For features that are only available with a time-based license, it is especially important that the license not expire before you can apply the new license. The ASA allows you to *stack* time-based licenses so that you do not have to worry about the license expiring or about losing time on your licenses because you installed the new one early.

When you install an identical time-based license as one already installed, then the licenses are combined, and the duration equals the combined duration.

For example:

1. You install a 52-week Botnet Traffic Filter license, and use the license for 25 weeks (27 weeks remain).
2. You then purchase another 52-week Botnet Traffic Filter license. When you install the second license, the licenses combine to have a duration of 79 weeks (52 weeks plus 27 weeks).

Similarly:

1. You install an 8-week 1000-session AnyConnect Premium license, and use it for 2 weeks (6 weeks remain).
2. You then install another 8-week 1000-session license, and the licenses combine to be 1000-sessions for 14 weeks (8 weeks plus 6 weeks).

If the licenses are not identical (for example, a 1000-session AnyConnect Premium license vs. a 2500-session license), then the licenses are *not* combined. Because only one time-based license per feature can be active, only one of the licenses can be active.

Although non-identical licenses do not combine, when the current license expires, the ASA automatically activates an installed license of the same feature if available.

Related Topics

Activate or Deactivate Keys, on page 65
Time-Based License Expiration, on page 52

Time-Based License Expiration

When the current license for a feature expires, the ASA automatically activates an installed license of the same feature if available. If there are no other time-based licenses available for the feature, then the permanent license is used.

If you have more than one additional time-based license installed for a feature, then the ASA uses the first license it finds; which license is used is not user-configurable and depends on internal operations. If you prefer to use a different time-based license than the one the ASA activated, then you must manually activate the license you prefer.

For example, you have a time-based 2500-session AnyConnect Premium license (active), a time-based 1000-session AnyConnect Premium license (inactive), and a permanent 500-session AnyConnect Premium license. While the 2500-session license expires, the ASA activates the 1000-session license. After the 1000-session license expires, the ASA uses the 500-session permanent license.

Related Topics

Activate or Deactivate Keys, on page 65

License Notes

The following sections include additional information about licenses.

AnyConnect Plus and Apex Licenses

The AnyConnect Plus or Apex license is a multi-use license that you can apply to multiple ASAs, all of which share a user pool as specified by the license. See https://www.cisco.com/go/license, and assign the PAK separately to each ASA. When you apply the resulting activation key to an ASA, it toggles on the VPN features to the maximum allowed, but the actual number of unique users across all ASAs sharing the license should not exceed the license limit. For more information, see:

- Cisco AnyConnect Ordering Guide
- AnyConnect Licensing Frequently Asked Questions (FAQ)

Other VPN License

Other VPN sessions include the following VPN types:

- IPsec remote access VPN using IKEv1
- IPsec site-to-site VPN using IKEv1
- IPsec site-to-site VPN using IKEv2

This license is included in the Base license.

Total VPN Sessions Combined, All Types

- Although the maximum VPN sessions add up to more than the maximum VPN AnyConnect and Other VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the ASA, so be sure to size your network appropriately.
- If you start a clientless SSL VPN session and then start an AnyConnect client session from the portal, 1 session is used in total. However, if you start the AnyConnect client first (from a standalone client, for example) and then log into the clientless SSL VPN portal, then 2 sessions are used.

VPN Load Balancing

VPN load balancing requires a Strong Encryption (3DES/AES) License.

Legacy VPN Licenses

Refer to the Supplemental end User License Agreement for AnyConnect for all relevant information on licensing.

Encryption License

The DES license cannot be disabled. If you have the 3DES license installed, DES is still available. To prevent the use of DES when you want to only use strong encryption, be sure to configure any relevant commands to use only strong encryption.

Total UC Proxy Sessions

Each TLS proxy session for Encrypted Voice Inspection is counted against the TLS license limit.

Other applications that use TLS proxy sessions do not count toward the TLS limit, for example, Mobility Advantage Proxy (which does not require a license).

Some applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command or in ASDM, using the **Configuration** > **Firewall** > **Unified Communications** > **TLS Proxy** pane. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a TLS proxy license that is higher than the default TLS proxy limit, the ASA automatically sets the TLS proxy limit to match the license. The TLS proxy limit takes precedence over the license limit; if you set the TLS proxy limit to be less than the license, then you cannot use all of the sessions in your license.

Note For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the license limit, then you see an error message to use the **tls-proxy maximum-sessions** command to raise the limit again (in ASDM, use the **TLS Proxy** pane). If you use failover and enter the **write standby** command or in ASDM, use **File > Save Running Configuration to Standby Unit** on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:

- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is no limit.

Note Only calls that require encryption/decryption for media are counted toward the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count toward the limit.

VLANs, Maximum

For an interface to count against the VLAN limit, you must assign a VLAN to it. For example:

```
interface gigabitethernet 0/0.100
vlan 100
```

Botnet Traffic Filter License

Requires a Strong Encryption (3DES/AES) License to download the dynamic database.

IPS Module License

The IPS module license lets you run the IPS software module on the ASA. You also need the IPS signature subscription on the IPS side.

See the following guidelines:

- To buy the IPS signature subscription you need to have the ASA with IPS pre-installed (the part number must include “IPS”, for example ASA5515-IPS-K9); you cannot buy the IPS signature subscription for a non-IPS part number ASA.

- For failover, you need the IPS signature subscription on both units; this subscription is not shared in failover, because it is not an ASA license.
- For failover, the IPS signature subscription requires a unique IPS module license per unit. Like other ASA licenses, the IPS module license is technically shared in the failover cluster license. However, because of the IPS signature subscription requirements, you must buy a separate IPS module license for each unit in failover.

Shared AnyConnect Premium Licenses (AnyConnect 3 and Earlier)

Note The shared license feature on the ASA is not supported with AnyConnect 4 and later licensing. AnyConnect licenses are shared and no longer require a shared server or participant license.

A shared license lets you purchase a large number of AnyConnect Premium sessions and share the sessions as needed among a group of ASAs by configuring one of the ASAs as a shared licensing server, and the rest as shared licensing participants.

Failover or ASA Cluster Licenses

With some exceptions, failover and cluster units do not require the same license on each unit. For earlier versions, see the licensing document for your version.

Failover License Requirements and Exceptions

Failover units do not require the same license on each unit. If you have licenses on both units, they combine into a single running failover cluster license. There are some exceptions to this rule. See the following table for precise licensing requirements for failover.

Model	License Requirement
ASA 5506-X and ASA 5506W-X	• Active/Standby—Security Plus License. • Active/Active—No Support. **Note** Each unit must have the same encryption license.

Model	License Requirement
ASA 5512-X through ASA 5555-X	• ASA 5512-X—Security Plus License. • Other models—Base License. **Note** • Each unit must have the same encryption license. • In multiple context mode, each unit must have the the same AnyConnect Apex license. • Each unit must have the same IPS module license. You also need the IPS signature subscription on the IPS side for both units. See the following guidelines: • To buy the IPS signature subscription you need to have the ASA with IPS pre-installed (the part number must include "IPS", for example ASA5525-IPS-K9); you cannot buy the IPS signature subscription for a non-IPS part number ASA. • You need the IPS signature subscription on both units; this subscription is not shared in failover, because it is not an ASA license. • The IPS signature subscription requires a unique IPS module license per unit. Like other ASA licenses, the IPS module license is technically shared in the failover cluster license. However, because of the IPS signature subscription requirements, you must buy a separate IPS module license for each unit in.
ASAv	See Failover Licenses for the ASAv, on page 113.
Firepower 9300	See Failover Licenses for the ASA on the Firepower 9300 Chassis, on page 113.
All other models	Base License or Standard License. **Note** • Each unit must have the same encryption license. • In multiple context mode, each unit must have the the same AnyConnect Apex license.

Note A valid permanent key is required; in rare instances, your PAK authentication key can be removed. If your key consists of all 0's, then you need to reinstall a valid authentication key before failover can be enabled.

ASA Cluster License Requirements and Exceptions

Cluster units do not require the same license on each unit. Typically, you buy a license only for the control unit; data units inherit the control unit license. If you have licenses on multiple units, they combine into a single running ASA cluster license.

There are exceptions to this rule. See the following table for precise licensing requirements for clustering.

Model	License Requirement
ASA 5585-X	Cluster License, supports up to 16 units. **Note** Each unit must have the same encryption license; each unit must have the same 10 GE I/O/Security Plus license (ASA 5585-X with SSP-10 and -20).
ASA 5512-X	Security Plus license, supports 2 units. **Note** Each unit must have the same encryption license.
ASA 5515-X, ASA 5525-X, ASA 5545-X, ASA 5555-X	Base License, supports 2 units. **Note** Each unit must have the same encryption license.
Firepower 9300 Chassis	See ASA Cluster Licenses for the ASA on the Firepower 9300 Chassis, on page 114.
All other models	No support.

How Failover or ASA Cluster Licenses Combine

For failover pairs or ASA clusters, the licenses on each unit are combined into a single running cluster license. If you buy separate licenses for each unit, then the combined license uses the following rules:

- For licenses that have numerical tiers, such as the number of sessions, the values from each unit's licenses are combined up to the platform limit. If all licenses in use are time-based, then the licenses count down simultaneously.

  For example, for failover:

  - You have two ASAs with 10 TLS Proxy sessions installed on each; the licenses will be combined for a total of 20 TLS Proxy sessions.
  - You have an ASA 5545-X with 1000 TLS Proxy sessions, and another with 2000 sessions; because the platform limit is 2000, the combined license allows 2000 TLS Proxy sessions.
  - You have two ASA 5545-X ASAs, one with 20 contexts and the other with 10 contexts; the combined license allows 30 contexts. For Active/Active failover, the contexts are divided between the two units. One unit can use 18 contexts and the other unit can use 12 contexts, for example, for a total of 30.

  For example, for ASA clustering:

  - You have 2 ASA 5516-X ASAs with the default 2 contexts. Because the platform limit is 5, the combined license allows a maximum of 4 contexts. Therefore, you can configure up to 4 contexts on the primary unit; each secondary unit will also have 4 contexts through configuration replication.
  - You have four ASA 5516-X ASAs, three units with 5 contexts each, and one unit with the default 2 contexts. Because the platform limit is 5, the licenses will be combined for a total of 5 contexts. Therefore, you can configure up to 5 contexts on the primary unit; each secondary unit will also have 5 contexts through configuration replication.

- For licenses that have a status of enabled or disabled, then the license with the enabled status is used.
- For time-based licenses that are enabled or disabled (and do not have numerical tiers), the duration is the combined duration of all licenses. The primary/control unit counts down its license first, and when it expires, the secondary/data unit(s) start counting down its license, and so on. This rule also applies to Active/Active failover and ASA clustering, even though all units are actively operating.

  For example, if you have 48 weeks left on the Botnet Traffic Filter license on two units, then the combined duration is 96 weeks.

Related Topics

Monitoring PAK Licenses, on page 89

Loss of Communication Between Failover or ASA Cluster Units

If the units lose communication for more than 30 days, then each unit reverts to the license installed locally. During the 30-day grace period, the combined running license continues to be used by all units.

If you restore communication during the 30-day grace period, then for time-based licenses, the time elapsed is subtracted from the primary/control license; if the primary/control license becomes expired, only then does the secondary/data license start to count down.

If you do not restore communication during the 30-day period, then for time-based licenses, time is subtracted from all unit licenses, if installed. They are treated as separate licenses and do not benefit from the combined license. The time elapsed includes the 30-day grace period.

For example:

1. You have a 52-week Botnet Traffic Filter license installed on two units. The combined running license allows a total duration of 104 weeks.
2. The units operate as a failover unit/ASA cluster for 10 weeks, leaving 94 weeks on the combined license (42 weeks on the primary/control, and 52 weeks on the secondary/data).
3. If the units lose communication (for example the primary/control unit fails), the secondary/data unit continues to use the combined license, and continues to count down from 94 weeks.
4. The time-based license behavior depends on when communication is restored:
   - Within 30 days—The time elapsed is subtracted from the primary/control unit license. In this case, communication is restored after 4 weeks. Therefore, 4 weeks are subtracted from the primary/control license leaving 90 weeks combined (38 weeks on the primary, and 52 weeks on the secondary).
   - After 30 days—The time elapsed is subtracted from both units. In this case, communication is restored after 6 weeks. Therefore, 6 weeks are subtracted from both the primary/control and secondary/data licenses, leaving 84 weeks combined (36 weeks on the primary/control, and 46 weeks on the secondary/data).

Upgrading Failover Pairs

Because failover pairs do not require the same license on both units, you can apply new licenses to each unit without any downtime. If you apply a permanent license that requires a reload, then you can fail over to the other unit while you reload. If both units require reloading, then you can reload them separately so that you have no downtime.

Related Topics

Activate or Deactivate Keys, on page 65

No Payload Encryption Models

You can purchase some models with No Payload Encryption. For export to some countries, payload encryption cannot be enabled on the Cisco ASA series. The ASA software senses a No Payload Encryption model, and disables the following features:

- Unified Communications
- VPN

You can still install the Strong Encryption (3DES/AES) license for use with management connections. For example, you can use ASDM HTTPS/SSL, SSHv2, Telnet and SNMPv3. You can also download the dynamic database for the Botnet Traffic Filter (which uses SSL).

When you view the license, VPN and Unified Communications licenses will not be listed.

Related Topics

Monitoring PAK Licenses, on page 89

Licenses FAQ

Can I activate multiple time-based licenses, for example, AnyConnect Premium and Botnet Traffic Filter?

Yes. You can use one time-based license per feature at a time.

Can I “stack” time-based licenses so that when the time limit runs out, it will automatically use the next license?

Yes. For identical licenses, the time limit is combined when you install multiple time-based licenses. For non-identical licenses (for example, a 1000-session AnyConnect Premium license and a 2500-session license), the ASA automatically activates the next time-based license it finds for the feature.

Can I install a new permanent license while maintaining an active time-based license?

Yes. Activating a permanent license does not affect time-based licenses.

For failover, can I use a shared licensing server as the primary unit, and the shared licensing backup server as the secondary unit?

No. The secondary unit has the same running license as the primary unit; in the case of the shared licensing server, they require a server license. The backup server requires a participant license. The backup server can be in a separate failover pair of two backup servers.

Do I need to buy the same licenses for the secondary unit in a failover pair?

No. Starting with Version 8.3(1), you do not have to have matching licenses on both units. Typically, you buy a license only for the primary unit; the secondary unit inherits the primary license when it becomes active. In the case where you also have a separate license on the secondary unit (for example, if you purchased matching licenses for pre-8.3 software), the licenses are combined into a running failover cluster license, up to the model limits.

Can I use a time-based or permanent AnyConnect Premium license in addition to a shared AnyConnect Premium license?

Yes. The shared license is used only after the sessions from the locally installed license (time-based or permanent) are used up.

Note On the shared licensing server, the permanent AnyConnect Premium license is not used; you can however use a time-based license at the same time as the shared licensing server license. In this case, the time-based license sessions are available for local AnyConnect Premium sessions only; they cannot be added to the shared licensing pool for use by participants.

Guidelines for PAK Licenses

Context Mode Guidelines

In multiple context mode, apply the activation key in the system execution space.

Failover Guidelines

See Failover or ASA Cluster Licenses, on page 55.

Model Guidelines

- Smart Licensing is supported on the ASAv only.
- Shared licenses are not supported on the ASAv, ASA 5506-X, ASA 5508-X, and ASA 5516-X.
- The ASA 5506-X and ASA 5506W-X do not support time-based licenses.

Upgrade and Downgrade Guidelines

Your activation key remains compatible if you upgrade to the latest version from any previous version. However, you might have issues if you want to maintain downgrade capability:

- Downgrading to Version 8.1 or earlier—After you upgrade, if you activate additional feature licenses that were introduced *before 8.2*, then the activation key continues to be compatible with earlier versions if you downgrade. However if you activate feature licenses that were introduced in *8.2 or later*, then the activation key is not backwards compatible. If you have an incompatible license key, then see the following guidelines:
    - If you previously entered an activation key in an earlier version, then the ASA uses that key (without any of the new licenses you activated in Version 8.2 or later).

    - If you have a new system and do not have an earlier activation key, then you need to request a new activation key compatible with the earlier version.

- Downgrading to Version 8.2 or earlier—Version 8.3 introduced more robust time-based key usage as well as failover license changes:
    - If you have more than one time-based activation key active, when you downgrade, only the most recently activated time-based key can be active. Any other keys are made inactive. If the last time-based license is for a feature introduced in 8.3, then that license still remains the active license even though it cannot be used in earlier versions. Reenter the permanent key or a valid time-based key.
    - If you have mismatched licenses on a failover pair, then downgrading will disable failover. Even if the keys are matching, the license used will no longer be a combined license.
    - If you have one time-based license installed, but it is for a feature introduced in 8.3, then after you downgrade, that time-based license remains active. You need to reenter the permanent key to disable the time-based license.

Additional Guidelines

- The activation key is not stored in your configuration file; it is stored as a hidden file in flash memory.
- The activation key is tied to the serial number of the device. Feature licenses cannot be transferred between devices (except in the case of a hardware failure). If you have to replace your device due to a hardware failure, and it is covered by Cisco TAC, contact the Cisco Licensing Team to have your existing license transferred to the new serial number. The Cisco Licensing Team will ask for the Product Authorization Key reference number and existing serial number.
- The serial number used for licensing is the one seen in the **show version** output. This serial number is different from the chassis serial number printed on the outside of your hardware. The chassis serial number is used for technical support, but not for licensing.
- Once purchased, you cannot return a license for a refund or for an upgraded license.
- On a single unit, you cannot add two separate licenses for the same feature together; for example, if you purchase a 25-session SSL VPN license, and later purchase a 50-session license, you cannot use 75 sessions; you can use a maximum of 50 sessions. (You may be able to purchase a larger license at an upgrade price, for example from 25 sessions to 75 sessions; this kind of upgrade should be distinguished from adding two separate licenses together).
- Although you can activate all license types, some features are incompatible with each other. In the case of the AnyConnect Essentials license, the license is incompatible with the following licenses: AnyConnect Premium license, shared AnyConnect Premium license, and Advanced Endpoint Assessment license. By default, if you install the AnyConnect Essentials license (if it is available for your model), it is used instead of the above licenses. You can disable the AnyConnect Essentials license in the configuration to restore use of the other licenses using the **webvpn**, and then the **no anyconnect-essentials** command.

Configure PAK Licenses

This section describes how to obtain an activation key and how to active it. You can also deactivate a key.

Order License PAKs and Obtain an Activation Key

To install a license on the ASA, you need Product Authorization Keys, which you can then register with Cisco.com to obtain an activation key. You can then enter the activation key on the ASA. You need a separate Product Authorization Key for each feature license. The PAKs are combined to give you a single activation key. You may have received all of your license PAKs in the box with your device. The ASA has the Base or Security Plus license pre-installed, along with the Strong Encryption (3DES/AES) license if you qualify for its use. If you need to manually request the Strong Encryption license (which is free), see http://www.cisco.com/go/license.

Before you begin

When you purchase 1 or more licenses for the device, you manage them in the Cisco Smart Software Manager:

https://software.cisco.com/#module/SmartLicensing

If you do not yet have an account, set up a new account. The Smart Software Manager lets you create a master account for your organization.

Procedure

Step 1 To purchase additional licenses, see http://www.cisco.com/go/ccw. See the following AnyConnect ordering guide and FAQ:

- Cisco AnyConnect Ordering Guide
- AnyConnect Licensing Frequently Asked Questions (FAQ)

After you order a license, you will then receive an email with a Product Authorization Key (PAK). For the AnyConnect licenses, you receive a multi-use PAK that you can apply to multiple ASAs that use the same pool of user sessions. The PAK email can take several days in some cases.

The ASA FirePOWER module uses a separate licensing mechanism from the ASA. See the quick start guide for your model for more information.

Step 2 Obtain the serial number for your ASA by entering the following command.

show version | grep Serial

The serial number used for licensing is different from the chassis serial number printed on the outside of your hardware. The chassis serial number is used for technical support, but not for licensing.

Step 3 To obtain the activation key, go to the following licensing website:

http://www.cisco.com/go/license

Step 4 Enter the following information, when prompted:

- Product Authorization Key (if you have multiple keys, enter one of the keys first. You have to enter each key as a separate process.)
- The serial number of your ASA
- Your e-mail address

An activation key is automatically generated and sent to the e-mail address that you provide. This key includes all features you have registered so far for permanent licenses. For time-based licenses, each license has a separate activation key.

Step 5 If you have additional Product Authorization Keys, repeat the process for each Product Authorization Key. After you enter all of the Product Authorization Keys, the final activation key provided includes all of the permanent features you registered.

Step 6 Install the activation key according to Activate or Deactivate Keys, on page 65.

Obtain a Strong Encryption License

To use ASDM (and many other features), you need to install the Strong Encryption (3DES/AES) license. If your ASA did not come with the Strong Encryption license pre-installed, you can request one for free. You must qualify for a Strong Encryption license based on your country.

Procedure

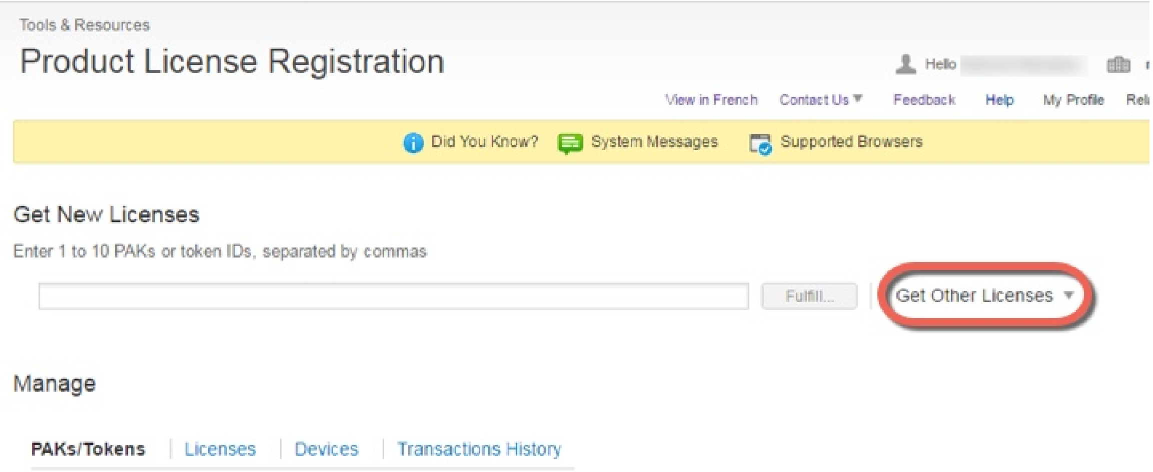
Step 1 Obtain the serial number for your ASA by entering the following command:

show version | grep Serial

This serial number is different from the chassis serial number printed on the outside of your hardware. The chassis serial number is used for technical support, but not for licensing.

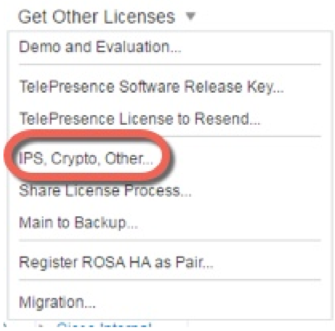
Step 2 See https://www.cisco.com/go/license, and click **Get Other Licenses**.

Figure 1: Get Other Licenses



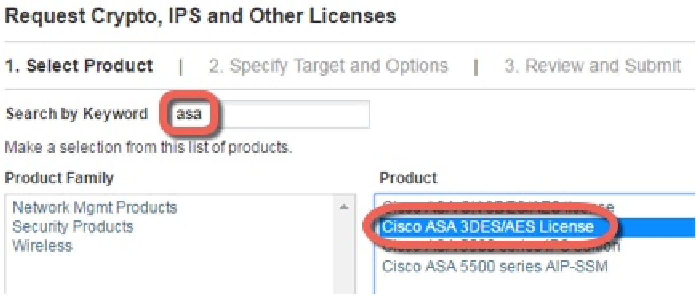
Step 3 Choose **IPS, Crypto, Other**.

Figure 2: IPS, Crypto, Other



Step 4 In the **Search by Keyword** field, enter **asa**, and select **Cisco ASA 3DES/AES License**.

Figure 3: Cisco ASA 3DES/AES License


Request Crypto, IPS and Other Licenses

1. Select Product | 2. Specify Target and Options | 3. Review and Submit

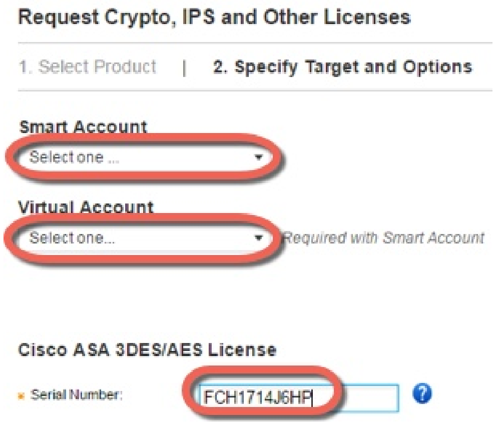
Search by Keyword: asa

Make a selection from this list of products.

Product Family
Network Mgmt Products
Security Products
Wireless
Product
Cisco ASA 3DES/AES License
Cisco ASA 5500 series AIP-SSM

Step 5 Select your **Smart Account**, **Virtual Account**, enter the ASA **Serial Number**, and click **Next**.

Figure 4: Smart Account, Virtual Account, and Serial Number


Request Crypto, IPS and Other Licenses

Smart Account: Select one ...

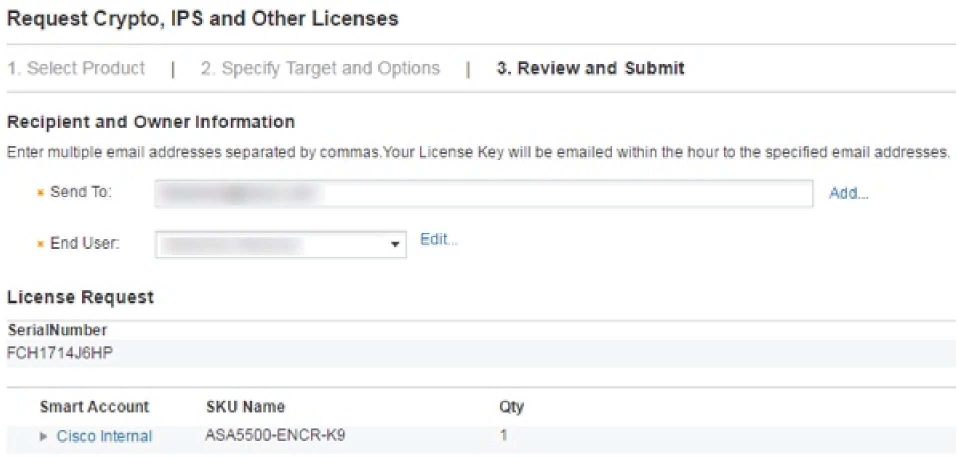
Virtual Account: Select one... Required with Smart Account

Cisco ASA 3DES/AES License

* Serial Number: FCH1714J6HP

Step 6 Your Send To email address and End User name are auto-filled; enter additional email addresses if needed. Check the **I Agree** check box, and click **Submit**.

Figure 5: Submit



Request Crypto, IPS and Other Licenses

1. Select Product | 2. Specify Target and Options | **3. Review and Submit**

Recipient and Owner Information

Enter multiple email addresses separated by commas.Your License Key will be emailed within the hour to the specified email addresses.

* Send To: [redacted] Add...

* End User: [redacted] Edit...

License Request

SerialNumber
FCH1714J6HP

Smart Account	SKU Name	Qty
▸ Cisco Internal	ASA5500-ENCR-K9	1

Step 7 You will then receive an email with the activation key, but you can also download the key right away from the **Manage** > **Licenses** area.

Step 8 Apply the activation key according to Activate or Deactivate Keys, on page 65.

Activate or Deactivate Keys

This section describes how to enter a new activation key, and how to activate and deactivate time-based keys.

Before you begin

- If you are already in multiple context mode, enter the activation key in the system execution space.
- Some permanent licenses require you to reload the ASA after you activate them. The following table lists the licenses that require reloading.

Table 2: Permanent License Reloading Requirements

Model	License Action Requiring Reload
All models	Downgrading the Encryption license.

Procedure

Step 1 Apply an activation key to the ASA:

activation-key *key* [**activate** | **deactivate**]

Example:

```
ciscoasa# activation-key 0xd11b3d48 0xa80a4c0a 0x48e0fd1c 0xb0443480 0x843fc490
```

The *key* is a five-element hexadecimal string with one space between each element. The leading 0x specifier is optional; all values are assumed to be hexadecimal.

You can install one permanent key, and multiple time-based keys. If you enter a new permanent key, it overwrites the already installed one.

The **activate** and **deactivate** keywords are available for time-based keys only. If you do not enter any value, **activate** is the default. The last time-based key that you activate for a given feature is the active one. To deactivate any active time-based key, enter the **deactivate** keyword. If you enter a key for the first time, and specify **deactivate**, then the key is installed on the ASA in an inactive state.

Step 2 (Might be required.) Reload the ASA:

reload

Some permanent licenses require you to reload the ASA after entering the new activation key. If you need to reload, you will see the following message:

```
WARNING: The running activation key was not updated with the requested key.
The flash activation key was updated with the requested key, and will become
active after the next reload.
```

Related Topics

Time-Based Licenses, on page 50

Configure a Shared License (AnyConnect 3 and Earlier)

Note The shared license feature on the ASA is not supported with AnyConnect 4 and later licensing. AnyConnect licenses are shared and no longer require a shared server or participant license.

This section describes how to configure the shared licensing server and participants.

About Shared Licenses

A shared license lets you purchase a large number of AnyConnect Premium sessions and share the sessions as needed among a group of ASAs by configuring one of the ASAs as a shared licensing server, and the rest as shared licensing participants.

About the Shared Licensing Server and Participants

The following steps describe how shared licenses operate:

1. Decide which ASA should be the shared licensing server, and purchase the shared licensing server license using that device serial number.
2. Decide which ASAs should be shared licensing participants, including the shared licensing backup server, and obtain a shared licensing participant license for each device, using each device serial number.

3. (Optional) Designate a second ASA as a shared licensing backup server. You can only specify one backup server.

Note The shared licensing backup server only needs a participant license.

4. Configure a shared secret on the shared licensing server; any participants with the shared secret can use the shared license.

5. When you configure the ASA as a participant, it registers with the shared licensing server by sending information about itself, including the local license and model information.

Note The participant needs to be able to communicate with the server over the IP network; it does not have to be on the same subnet.

6. The shared licensing server responds with information about how often the participant should poll the server.

7. When a participant uses up the sessions of the local license, it sends a request to the shared licensing server for additional sessions in 50-session increments.

8. The shared licensing server responds with a shared license. The total sessions used by a participant cannot exceed the maximum sessions for the platform model.

Note The shared licensing server can also participate in the shared license pool. It does not need a participant license as well as the server license to participate.

   a. If there are not enough sessions left in the shared license pool for the participant, then the server responds with as many sessions as available.

   b. The participant continues to send refresh messages requesting more sessions until the server can adequately fulfill the request.

9. When the load is reduced on a participant, it sends a message to the server to release the shared sessions.

Note The ASA uses SSL between the server and participant to encrypt all communications.

Communication Issues Between Participant and Server

See the following guidelines for communication issues between the participant and server:

- If a participant fails to send a refresh after 3 times the refresh interval, then the server releases the sessions back into the shared license pool.
- If the participant cannot reach the license server to send the refresh, then the participant can continue to use the shared license it received from the server for up to 24 hours.

- If the participant is still not able to communicate with a license server after 24 hours, then the participant releases the shared license, even if it still needs the sessions. The participant leaves existing connections established, but cannot accept new connections beyond the license limit.
- If a participant reconnects with the server before 24 hours expires, but after the server expired the participant sessions, then the participant needs to send a new request for the sessions; the server responds with as many sessions as can be reassigned to that participant.

About the Shared Licensing Backup Server

The shared licensing backup server must register successfully with the main shared licensing server before it can take on the backup role. When it registers, the main shared licensing server syncs server settings as well as the shared license information with the backup, including a list of registered participants and the current license usage. The main server and backup server sync the data at 10 second intervals. After the initial sync, the backup server can successfully perform backup duties, even after a reload.

When the main server goes down, the backup server takes over server operation. The backup server can operate for up to 30 continuous days, after which the backup server stops issuing sessions to participants, and existing sessions time out. Be sure to reinstate the main server within that 30-day period. Critical-level syslog messages are sent at 15 days, and again at 30 days.

When the main server comes back up, it syncs with the backup server, and then takes over server operation.

When the backup server is not active, it acts as a regular participant of the main shared licensing server.

Note When you first launch the main shared licensing server, the backup server can only operate independently for 5 days. The operational limit increases day-by-day, until 30 days is reached. Also, if the main server later goes down for any length of time, the backup server operational limit decrements day-by-day. When the main server comes back up, the backup server starts to increment again day-by-day. For example, if the main server is down for 20 days, with the backup server active during that time, then the backup server will only have a 10-day limit left over. The backup server “recharges” up to the maximum 30 days after 20 more days as an inactive backup. This recharging function is implemented to discourage misuse of the shared license.

Failover and Shared Licenses

This section describes how shared licenses interact with failover.

Failover and Shared License Servers

This section describes how the main server and backup server interact with failover. Because the shared licensing server is also performing normal duties as the ASA, including performing functions such as being a VPN gateway and firewall, then you might need to configure failover for the main and backup shared licensing servers for increased reliability.

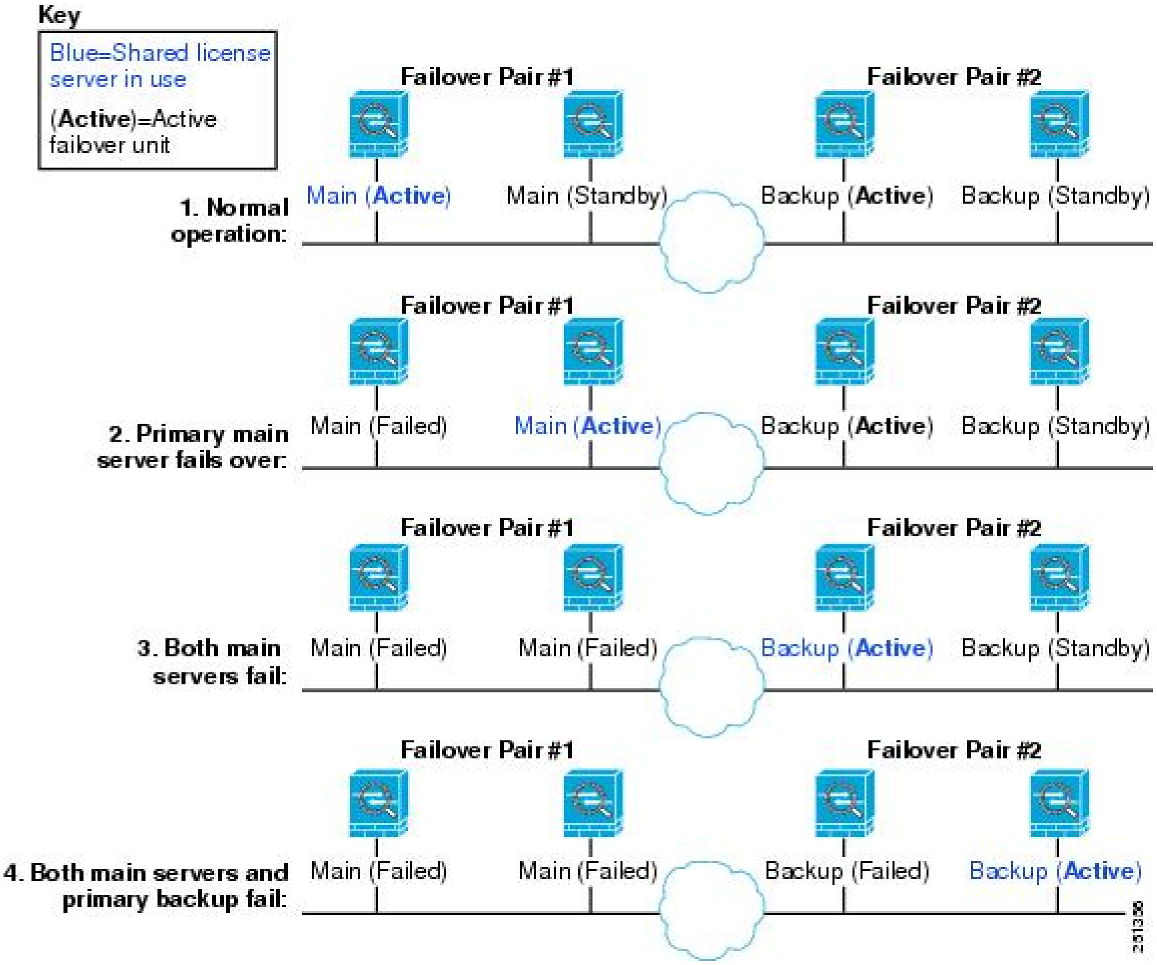
Note The backup server mechanism is separate from, but compatible with, failover.

Shared licenses are supported only in single context mode, so Active/Active failover is not supported.

For Active/Standby failover, the primary unit acts as the main shared licensing server, and the standby unit acts as the main shared licensing server after failover. The standby unit does *not* act as the backup shared licensing server. Instead, you can have a second pair of units acting as the backup server, if desired.

For example, you have a network with 2 failover pairs. Pair #1 includes the main licensing server. Pair #2 includes the backup server. When the primary unit from Pair #1 goes down, the standby unit immediately becomes the new main licensing server. The backup server from Pair #2 never gets used. Only if both units in Pair #1 go down does the backup server in Pair #2 come into use as the shared licensing server. If Pair #1 remains down, and the primary unit in Pair #2 goes down, then the standby unit in Pair #2 comes into use as the shared licensing server (see the following figure).

Figure 6: Failover and Shared License Servers



The standby backup server shares the same operating limits as the primary backup server; if the standby unit becomes active, it continues counting down where the primary unit left off.

Related Topics

About the Shared Licensing Backup Server, on page 68

Failover and Shared License Participants

For participant pairs, both units register with the shared licensing server using separate participant IDs. The active unit syncs its participant ID with the standby unit. The standby unit uses this ID to generate a transfer request when it switches to the active role. This transfer request is used to move the shared sessions from the previously active unit to the new active unit.

Maximum Number of Participants

The ASA does not limit the number of participants for the shared license; however, a very large shared network could potentially affect the performance on the licensing server. In this case, you can increase the delay between participant refreshes, or you can create two shared networks.

Configure the Shared Licensing Server

This section describes how to configure the ASA to be a shared licensing server.

Before you begin

The server must have a shared licensing server key.

Procedure

Step 1 Set the shared secret:

license-server secret *secret*

Example:

```
ciscoasa(config)# license-server secret farscape
```

The *secret* is a string between 4 and 128 ASCII characters. Any participant with this secret can use the licensing server.

Step 2 (Optional) Set the refresh interval:

license-server refresh-interval *seconds*

Example:

```
ciscoasa(config)# license-server refresh-interval 100
```

The interval is between 10 and 300 seconds; this value is provided to participants to set how often they should communicate with the server. The default is 30 seconds.

Step 3 (Optional) Set the port on which the server listens for SSL connections from participants:

license-server port *port*

Example:

```
ciscoasa(config)# license-server port 40000
```

The *port* is between 1 and 65535. The default is TCP port 50554.

Step 4 (Optional) Identify the backup server IP address and serial number:

license-server backup *address* **backup-id** *serial_number* [**ha-backup-id** *ha_serial_number*]

Example:

```
ciscoasa(config)# license-server backup 10.1.1.2 backup-id JMX0916L0Z4 ha-backup-id
JMX1378N0W3
```

If the backup server is part of a failover pair, identify the standby unit serial number as well. You can only identify 1 backup server and its optional standby unit.

Step 5 Enable this unit to be the shared licensing server:

license-server enable *interface_name*

Example:

```
ciscoasa(config)# license-server enable inside
```

Specify the interface on which participants contact the server. You can repeat this command for as many interfaces as desired.

Examples

The following example sets the shared secret, changes the refresh interval and port, configures a backup server, and enables this unit as the shared licensing server on the inside interface and dmz interface:

```
ciscoasa(config)# license-server secret farscape
ciscoasa(config)# license-server refresh-interval 100
ciscoasa(config)# license-server port 40000
ciscoasa(config)# license-server backup 10.1.1.2 backup-id JMX0916L0Z4 ha-backup-id
JMX1378N0W3
ciscoasa(config)# license-server enable inside
ciscoasa(config)# license-server enable dmz
```

Configure the Shared Licensing Backup Server (Optional)

This section enables a shared license participant to act as the backup server if the main server goes down.

Before you begin

The backup server must have a shared licensing participant key.

Procedure

Step 1 Identify the shared licensing server IP address and shared secret:

license-server address *address* **secret** *secret* [**port** *port*]

Example:

```
ciscoasa(config)# license-server address 10.1.1.1 secret farscape
```

If you changed the default port in the server configuration, set the port for the backup server to match.

Step 2 Enable this unit to be the shared licensing backup server:

license-server backup enable *interface_name*

Example:

```
ciscoasa(config)# license-server backup enable inside
```

Specify the interface on which participants contact the server. You can repeat this command for as many interfaces as desired.

Examples

The following example identifies the license server and shared secret, and enables this unit as the backup shared license server on the inside interface and dmz interface:

```
ciscoasa(config)# license-server address 10.1.1.1 secret farscape
ciscoasa(config)# license-server backup enable inside
ciscoasa(config)# license-server backup enable dmz
```

Configure the Shared Licensing Participant

This section configures a shared licensing participant to communicate with the shared licensing server.

Before you begin

The participant must have a shared licensing participant key.

Procedure

Step 1 Identify the shared licensing server IP address and shared secret:

license-server address *address* **secret** *secret* [**port** *port*]

Example:

```
ciscoasa(config)# license-server address 10.1.1.1 secret farscape
```

If you changed the default port in the server configuration, set the port for the participant to match.

Step 2 (Optional) If you configured a backup server, enter the backup server address:

license-server backup address *address*

Example:

```
ciscoasa(config)# license-server backup address 10.1.1.2
```

Examples

The following example sets the license server IP address and shared secret, as well as the backup license server IP address:

```
ciscoasa(config)# license-server address 10.1.1.1 secret farscape
ciscoasa(config)# license-server backup address 10.1.1.2
```

Supported Feature Licenses Per Model

This section describes the licenses available for each model as well as important notes about licenses.

Licenses Per Model

This section lists the feature licenses available for each model:

Items that are in *italics* are separate, optional licenses that can replace the Base (or Security Plus, and so on) license version. You can mix and match optional licenses.

Note Some features are incompatible with each other. See the individual feature chapters for compatibility information.

If you have a No Payload Encryption model, then some of the features below are not supported. See No Payload Encryption Models, on page 59 for a list of unsupported features.

For detailed information about licenses, see License Notes, on page 52.

ASA 5506-X and ASA 5506W-X License Features

The following table shows the licensed features for the ASA 5506-X and ASA 5506W-X.

Licenses	Base License	Security Plus License
Firewall Licenses		
Botnet Traffic Filter	No support	No Support **Note** The licensing information might show that this license is Enabled. You should ignore the display, as this license is not supported.

Licenses	Base License		Security Plus License	
Firewall Conns, Concurrent	20,000		50,000	
GTP/GPRS	No Support		No Support	
Total UC Proxy Sessions	160		160	
VPN Licenses				
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 50 maximum*	Disabled	*Optional AnyConnect Plus or Apex license: 50 maximum*
Other VPN Peers	10		50	
Total VPN Peers, combined all types	50		50	
VPN Load Balancing	No support		No support	
General Licenses				
Encryption	Base (DES)	*Opt. lic.: Strong (3DES/AES)*	Base (DES)	*Opt. lic.: Strong (3DES/AES)*
Failover	No support		Active/Standby	
Security Contexts	No support		No support	
Clustering	No Support		No Support	
VLANs, Maximum	5		30	

ASA 5506H-X License Features

The following table shows the licensed features for the ASA 5506H-X.

Licenses	Base License
Firewall Licenses	

Licenses	Base License	
Botnet Traffic Filter	No Support **Note** The licensing information might show that this license is Enabled. You should ignore the display, as this license is not supported.	
Firewall Conns, Concurrent	50,000	
GTP/GPRS	No Support	
Total UC Proxy Sessions	160	
VPN Licenses		
AnyConnect Plus or Apex license (purchased separately), maximum premium peers	50	
Total VPN Peers, combined all types	50	
Other VPN Peers	50	
VPN Load Balancing	Enabled	
General Licenses		
Encryption	Base (DES)	*Opt. lic.: Strong (3DES/AES)*
Failover	Active/Standby or Active/Active	
Security Contexts	No Support	
Clustering	No Support	
VLANs, Maximum	30	

ASA 5508-X License Features

The following table shows the licensed features for the ASA 5508-X.

Licenses	Base License
Firewall Licenses	

Licenses	Base License		
Botnet Traffic Filter	No Support **Note** The licensing information might show that this license is Enabled. You should ignore the display, as this license is not supported.		
Firewall Conns, Concurrent	100,000		
GTP/GPRS	No Support		
Total UC Proxy Sessions	320		
VPN Licenses			
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 100 maximum*	
Total VPN Peers, combined all types	100		
Other VPN Peers	100		
VPN Load Balancing	Enabled		
General Licenses			
Encryption	Base (DES)	*Opt. lic.: Strong (3DES/AES)*	
Failover	Active/Standby or Active/Active		
Security Contexts	2	*Optional licenses:*	*5*
Clustering	No Support		
VLANs, Maximum	50		

ASA 5512-X License Features

The following table shows the licensed features for the ASA 5512-X.

Licenses	Base License		Security Plus License	
Firewall Licenses				
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*	Disabled	*Optional Time-based license: Available*

Licenses	Base License						Security Plus License					
Firewall Conns, Concurrent	100,000						250,000					
GTP/GPRS	No support						No Support					
Total UC Proxy Sessions	2	*Optional licenses:*					2	*Optional licenses:*				
		24	*50*	*100*	*250*	*500*		*24*	*50*	*100*	*250*	*500*
VPN Licenses												
AnyConnect peers	Disabled		*Optional AnyConnect Plus or Apex license: 250 maximum*				Disabled		*Optional AnyConnect Plus or Apex license: 250 maximum*			
Other VPN Peers	250						250					
Total VPN Peers, combined all types	250						250					
VPN Load Balancing	No support						Enabled					
General Licenses												
Encryption	Base (DES)		*Opt. lic.: Strong (3DES/AES)*				Base (DES)		*Opt. lic.: Strong (3DES/AES)*			
Failover	No support						Active/Standby or Active/Active					
Security Contexts	No support						2	*Optional licenses:*			*5*	
Clustering	No Support						2					
IPS Module	Disabled		*Optional license: Available*				Disabled		*Optional license: Available*			
VLANs, Maximum	50						100					

ASA 5515-X License Features

The following table shows the licensed features for the ASA 5515-X.

Licenses	Base License						
Firewall Licenses							
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*					
Firewall Conns, Concurrent	250,000						
GTP/GPRS	No Support						
Total UC Proxy Sessions	2	*Optional licenses:*	*24*	*50*	*100*	*250*	*500*
VPN Licenses							
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 250 maximum*					
Other VPN Peers	250						
Total VPN Peers, combined all types	250						
VPN Load Balancing	Enabled						
General Licenses							
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*					
Failover	Active/Standby or Active/Active						
Security Contexts	2	*Optional licenses:*	*5*				
Clustering	2						
IPS Module	Disabled	*Optional license: Available*					
VLANs, Maximum	100						

ASA 5516-X License Features

The following table shows the licensed features for the ASA 5516-X.

Licenses	Base License		
Firewall Licenses			
Botnet Traffic Filter	No Support **Note** The licensing information might show that this license is Enabled. You should ignore the display, as this license is not supported.		
Firewall Conns, Concurrent	250,000		
GTP/GPRS	No Support		
Total UC Proxy Sessions	1000		
VPN Licenses			
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 300 maximum*	
Other VPN Peers	300		
Total VPN Peers, combined all types	300		
VPN Load Balancing	Enabled		
General Licenses			
Encryption	Base (DES)	*Opt. lic.: Strong (3DES/AES)*	
Failover	Active/Standby or Active/Active		
Security Contexts	2	*Optional licenses:*	*5*
Clustering	No Support		
VLANs, Maximum	150		

ASA 5525-X License Features

The following table shows the licensed features for the ASA 5525-X.

Licenses	Base License
Firewall Licenses	

Licenses	Base License								
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*							
Firewall Conns, Concurrent	500,000								
GTP/GPRS	Disabled	*Optional license: Available*							
Total UC Proxy Sessions	2	*Optional licenses:*	*24*	*50*	*100*	*250*	*500*	*750*	*1000*
VPN Licenses									
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 750 maximum*							
Other VPN Peers	750								
Total VPN Peers, combined all types	750								
VPN Load Balancing	Enabled								
General Licenses									
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*							
Failover	Active/Standby or Active/Active								
Security Contexts	2	*Optional licenses:*	*5*	*10*	*20*				
Clustering	2								
IPS Module	Disabled	*Optional license: Available*							
VLANs, Maximum	200								

ASA 5545-X License Features

The following table shows the licensed features for the ASA 5545-X.

Licenses	Base License									
Firewall Licenses										
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*								
Firewall Conns, Concurrent	750,000									
GTP/GPRS	Disabled	*Optional license: Available*								
Total UC Proxy Sessions	2	*Optional licenses:*	*24*	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*
VPN Licenses										
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 2500 maximum*								
Other VPN Peers	2500									
Total VPN Peers, combined all types	2500									
VPN Load Balancing	Enabled									
General Licenses										
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*								
Failover	Active/Standby or Active/Active									
Security Contexts	2	*Optional licenses:*	*5*	*10*	*20*	*50*				
Clustering	2									
IPS Module	Disabled	*Optional license: Available*								

Licenses	Base License
VLANs, Maximum	300

ASA 5555-X License Features

The following table shows the licensed features for the ASA 5555-X.

Licenses	Base License									
Firewall Licenses										
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*								
Firewall Conns, Concurrent	1,000,000									
GTP/GPRS	Disabled	*Optional license: Available*								
Total UC Proxy Sessions	2	*Optional licenses:*								
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*	*3000*
VPN Licenses										
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 5000 maximum*								
Other VPN Peers	5000									
Total VPN Peers, combined all types	5000									
VPN Load Balancing	Enabled									
General Licenses										
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*								
Failover	Active/Standby or Active/Active									

Licenses	Base License						
Security Contexts	2	*Optional licenses:*	*5*	*10*	*20*	*50*	*100*
Clustering	2						
IPS Module	Disabled	*Optional license: Available*					
VLANs, Maximum	500						

ASA 5585-X with SSP-10 License Features

The following table shows the licensed features for the ASA 5585-X with SSP-10.

You can use two SSPs of the same level in the same chassis. Mixed-level SSPs are not supported (for example, an SSP-10 with an SSP-20 is not supported). Each SSP acts as an independent device, with separate configurations and management. You can use the two SSPs as a failover pair if desired.

Licenses	Base and Security Plus Licenses									
Firewall Licenses										
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*								
Firewall Conns, Concurrent	1,000,000									
GTP/GPRS	Disabled	*Optional license: Available*								
Total UC Proxy Sessions	2	*Optional licenses:*								
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*	*3000*
VPN Licenses										
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 5000 maximum*								
Other VPN Peers	5000									
Total VPN Peers, combined all types	5000									

Licenses	Base and Security Plus Licenses						
VPN Load Balancing	Enabled						
General Licenses							
10 GE I/O	**Base License**: Disabled; fiber ifcs run at 1 GE	**Security Plus License**: Enabled; fiber ifcs run at 10 GE					
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*					
Failover	Active/Standby or Active/Active						
Security Contexts	2	*Optional licenses:*	*5*	*10*	*20*	*50*	*100*
Clustering	Disabled	*Optional license: Available for 16 units*					
VLANs, Maximum	1024						

ASA 5585-X with SSP-20 License Features

The following table shows the licensed features for the ASA 5585-X with SSP-20.

You can use two SSPs of the same level in the same chassis. Mixed-level SSPs are not supported (for example, an SSP-20 with an SSP-40 is not supported). Each SSP acts as an independent device, with separate configurations and management. You can use the two SSPs as a failover pair if desired.

Note With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

Licenses	Base and Security Plus Licenses	
Firewall Licenses		
Botnet Traffic Filter	Disabled	*Optional Time-based license: Available*
Firewall Conns, Concurrent	2,000,000	
GTP/GPRS	Disabled	*Optional license: Available*

Licenses	Base and Security Plus Licenses											
Total UC Proxy Sessions	2	*Optional licenses:*										
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*	*3000*	*5000*	*10,000*
VPN Licenses												
AnyConnect peers	Disabled		*Optional AnyConnect Plus or Apex license: 10,000 maximum*									
Other VPN Peers	10,000											
Total VPN Peers, combined all types	10,000											
VPN Load Balancing	Enabled											
General Licenses												
10 GE I/O	**Base License**: Disabled; fiber ifcs run at 1 GE						**Security Plus License**: Enabled; fiber ifcs run at 10 GE					
Encryption	Base (DES)		*Optional license: Strong (3DES/AES)*									
Failover	Active/Standby or Active/Active											
Security Contexts	2	*Optional licenses:*			*5*	*10*	*20*	*50*	*100*	*250*		
Clustering	Disabled		*Optional license: Available for 16 units*									
VLANs, Maximum	1024											

ASA 5585-X with SSP-40 and -60 License Features

The following table shows the licensed features for the ASA 5585-X with SSP-40 and -60.

You can use two SSPs of the same level in the same chassis. Mixed-level SSPs are not supported (for example, an SSP-40 with an SSP-60 is not supported). Each SSP acts as an independent device, with separate configurations and management. You can use the two SSPs as a failover pair if desired.

Note With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

Licenses	**Base License**											
Firewall Licenses												
Botnet Traffic Filter	Disabled		*Optional Time-based license: Available*									
Firewall Conns, Concurrent	**5585-X with SSP-40**: 4,000,000						**5585-X with SSP-60**: 10,000,000					
GTP/GPRS	Disabled		*Optional license: Available*									
Total UC Proxy Sessions	2	*Optional licenses:*										
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*	*3000*	*5000*	*10,000*
VPN Licenses												
AnyConnect peers	Disabled		*Optional AnyConnect Plus or Apex license: 10,000 maximum*									
Other VPN Peers	10,000											
Total VPN Peers, combined all types	10,000											
VPN Load Balancing	Enabled											
General Licenses												
10 GE I/O	Enabled; fiber ifcs run at 10 GE											
Encryption	Base (DES)		*Optional license: Strong (3DES/AES)*									
Failover	Active/Standby or Active/Active											
Security Contexts	2	*Optional licenses:*			*5*	*10*	*20*	*50*	*100*	*250*		

Licenses	Base License	
Clustering	Disabled	*Optional license: Available for 16 units*
VLANs, Maximum	1024	

ASASM License Features

The following table shows the licensed features for the ASA Services Module.

Note With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

<table>
<tr><th>Licenses</th><th colspan="12">Base License</th></tr>
<tr><td colspan="13"><b>Firewall Licenses</b></td></tr>
<tr><td>Botnet Traffic Filter</td><td colspan="2">Disabled</td><td colspan="10"><i>Optional Time-based license: Available</i></td></tr>
<tr><td>Firewall Conns, Concurrent</td><td colspan="12">10,000,000</td></tr>
<tr><td>GTP/GPRS</td><td colspan="2">Disabled</td><td colspan="10"><i>Optional license: Available</i></td></tr>
<tr><td rowspan="2">Total UC Proxy Sessions</td><td rowspan="2">2</td><td colspan="11"><i>Optional licenses:</i></td></tr>
<tr><td><i>24</i></td><td><i>50</i></td><td><i>100</i></td><td><i>250</i></td><td><i>500</i></td><td><i>750</i></td><td><i>1000</i></td><td><i>2000</i></td><td><i>3000</i></td><td><i>5000</i></td><td><i>10,000</i></td></tr>
<tr><td colspan="13"><b>VPN Licenses</b></td></tr>
<tr><td>AnyConnect peers</td><td colspan="2">Disabled</td><td colspan="10"><i>Optional AnyConnect Plus or Apex license: 10,000 maximum</i></td></tr>
<tr><td>Other VPN Peers</td><td colspan="12">10,000</td></tr>
<tr><td>Total VPN Peers, combined all types</td><td colspan="12">10,000</td></tr>
</table>

Licenses	Base License						
VPN Load Balancing	Enabled						
General Licenses							
Encryption	Base (DES)		*Optional license: Strong (3DES/AES)*				
Failover	Active/Standby or Active/Active						
Security Contexts	2	*Optional licenses:*					
		5	*10*	*20*	*50*	*100*	*250*
Clustering	No support						
VLANs, Maximum	1000						

ISA 3000 License Features

The following table shows the licensed features for the ISA 3000.

Licenses	Base License		Security Plus License	
Firewall Licenses				
Botnet Traffic Filter	No support		No Support	
Firewall Conns, Concurrent	20,000		50,000	
GTP/GPRS	No Support		No Support	
Total UC Proxy Sessions	160		160	
VPN Licenses				
AnyConnect peers	Disabled	*Optional AnyConnect Plus or Apex license: 25 maximum*	Disabled	*Optional AnyConnect Plus or Apex license: 25 maximum*
Other VPN Peers	10		50	

Licenses	Base License		Security Plus License	
Total VPN Peers, combined all types	25		50	
VPN Load Balancing	No support		No support	
General Licenses				
Encryption	Base (DES)	*Opt. lic.: Strong (3DES/AES)*	Base (DES)	*Opt. lic.: Strong (3DES/AES)*
Failover	No support		Active/Standby	
Security Contexts	No support		No Support	
Clustering	No Support		No Support	
VLANs, Maximum	5		25	

Monitoring PAK Licenses

This section describes how to view license information.

Viewing Your Current License

This section describes how to view your current license, and for time-based activation keys, how much time the license has left.

Before you begin

If you have a No Payload Encryption model, then you view the license, VPN and Unified Communications licenses will not be listed. See No Payload Encryption Models, on page 59 for more information.

Procedure

Show the permanent license, active time-based licenses, and the running license, which is a combination of the permanent license and active time-based licenses:

show activation-key [**detail**]

The **detail** keyword also shows inactive time-based licenses.

For failover or cluster units, this command also shows the “cluster” license, which is the combined keys of all units.

Examples

Example 1: Standalone Unit Output for the show activation-key command

The following is sample output from the **show activation-key** command for a standalone unit that shows the running license (the combined permanent license and time-based licenses), as well as each active time-based license:

```
ciscoasa# show activation-key

Serial Number:  JMX1232L11M
Running Permanent Activation Key: 0xce06dc6b 0x8a7b5ab7 0xa1e21dd4 0xd2c4b8b8 0xc4594f9c
Running Timebased Activation Key: 0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285
Running Timebased Activation Key: 0xyadayad2 0xyadayad2 0xyadayad2 0xyadayad2 0xyadayad2

Licensed features for this platform:
Maximum Physical Interfaces       : Unlimited      perpetual
Maximum VLANs                     : 150            perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
VPN-DES                           : Enabled        perpetual
VPN-3DES-AES                      : Enabled        perpetual
Security Contexts                 : 10             perpetual
GTP/GPRS                          : Enabled        perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Enabled        perpetual
  Shared AnyConnect Premium Peers : 12000          perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions           : 12             62 days
Total UC Proxy Sessions           : 12             62 days
Botnet Traffic Filter             : Enabled        646 days
Intercompany Media Engine         : Disabled       perpetual

This platform has a Base license.

The flash permanent activation key is the SAME as the running permanent key.

Active Timebased Activation Key:
0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285
Botnet Traffic Filter        : Enabled    646 days

0xyadayad2 0xyadayad2 0xyadayad2 0xyadayad2 0xyadayad2
Total UC Proxy Sessions      : 10         62 days
```

Example 2: Standalone Unit Output for show activation-key detail

The following is sample output from the **show activation-key detail** command for a standalone unit that shows the running license (the combined permanent license and time-based licenses), as well as the permanent license and each installed time-based license (active and inactive):

```
ciscoasa# show activation-key detail

Serial Number:  88810093382
Running Permanent Activation Key: 0xce06dc6b 0x8a7b5ab7 0xa1e21dd4 0xd2c4b8b8 0xc4594f9c
Running Timebased Activation Key: 0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285

Licensed features for this platform:
Maximum Physical Interfaces    : 8              perpetual
VLANs                          : 20             DMZ Unrestricted
Dual ISPs                      : Enabled        perpetual
VLAN Trunk Ports               : 8              perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Standby perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Enabled        perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 25             perpetual
Total VPN Peers                   : 25              perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Enabled        39 days
Intercompany Media Engine      : Disabled       perpetual

This platform has an ASA 5512-X Security Plus license.

Running Permanent Activation Key: 0xce06dc6b 0x8a7b5ab7 0xa1e21dd4 0xd2c4b8b8 0xc4594f9c

Licensed features for this platform:
Maximum Physical Interfaces    : 8              perpetual
VLANs                          : 20             DMZ Unrestricted
Dual ISPs                      : Enabled        perpetual
VLAN Trunk Ports               : 8              perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Standby perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Enabled        perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 25             perpetual
Total VPN Peers                   : 25              perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Enabled        39 days
Intercompany Media Engine      : Disabled       perpetual

The flash permanent activation key is the SAME as the running permanent key.

Active Timebased Activation Key:
0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285
Botnet Traffic Filter           : Enabled    39 days

Inactive Timebased Activation Key:
0xyadayada3 0xyadayada3 0xyadayada3 0xyadayada3 0xyadayada3
AnyConnect Premium Peers                   : 25     7 days
```

Example 3: Primary Unit Output in a Failover Pair for show activation-key detail

The following is sample output from the **show activation-key detail** command for the primary failover unit that shows:

- The primary unit license (the combined permanent license and time-based licenses).
- The “Failover Cluster” license, which is the combined licenses from the primary and secondary units. This is the license that is actually running on the ASA. The values in this license that reflect the combination of the primary and secondary licenses are in bold.
- The primary unit permanent license.
- The primary unit installed time-based licenses (active and inactive).

```
ciscoasa# show activation-key detail

Serial Number:  P3000000171
Running Permanent Activation Key: 0xce06dc6b 0x8a7b5ab7 0xa1e21dd4 0xd2c4b8b8 0xc4594f9c
Running Timebased Activation Key: 0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285

Licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Enabled        perpetual
Security Contexts              : 12             perpetual
GTP/GPRS                       : Enabled        perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled        perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Enabled        33 days
Intercompany Media Engine      : Disabled       perpetual

This platform has an ASA 5520 VPN Plus license.

Failover cluster licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Enabled        perpetual
Security Contexts              : 12             perpetual
GTP/GPRS                       : Enabled        perpetual
AnyConnect Premium Peers       : 4           perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled       perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
```

```
UC Phone Proxy Sessions        : 4              perpetual
Total UC Proxy Sessions        : 4              perpetual
Botnet Traffic Filter          : Enabled        33 days
Intercompany Media Engine      : Disabled       perpetual

This platform has an ASA 5520 VPN Plus license.

Running Permanent Activation Key: 0xce06dc6b 0x8a7b5ab7 0xa1e21dd4 0xd2c4b8b8 0xc4594f9c

Licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Disabled       perpetual
Security Contexts              : 2              perpetual
GTP/GPRS                       : Disabled       perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled        perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Disabled       perpetual
Intercompany Media Engine      : Disabled       perpetual

The flash permanent activation key is the SAME as the running permanent key.

Active Timebased Activation Key:
0xa821d549 0x35725fe4 0xc918b97b 0xce0b987b 0x47c7c285
Botnet Traffic Filter           : Enabled    33 days

Inactive Timebased Activation Key:
0xyadayad3 0xyadayad3 0xyadayad3 0xyadayad3 0xyadayad3
Security Contexts               : 2          7 days
AnyConnect Premium Peers                   : 100        7 days

0xyadayad4 0xyadayad4 0xyadayad4 0xyadayad4 0xyadayad4
Total UC Proxy Sessions         : 100        14 days
```

Example 4: Secondary Unit Output in a Failover Pair for show activation-key detail

The following is sample output from the **show activation-key detail** command for the secondary failover unit that shows:

- The secondary unit license (the combined permanent license and time-based licenses).
- The “Failover Cluster” license, which is the combined licenses from the primary and secondary units. This is the license that is actually running on the ASA. The values in this license that reflect the combination of the primary and secondary licenses are in bold.
- The secondary unit permanent license.
- The secondary installed time-based licenses (active and inactive). This unit does not have any time-based licenses, so none display in this sample output.

```
ciscoasa# show activation-key detail

Serial Number:  P3000000011
Running Activation Key: 0xyadayad1 0xyadayad1 0xyadayad1 0xyadayad1 0xyadayad1

Licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Disabled       perpetual
Security Contexts              : 2              perpetual
GTP/GPRS                       : Disabled       perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled        perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Disabled       perpetual
Intercompany Media Engine      : Disabled       perpetual

This platform has an ASA 5520 VPN Plus license.

Failover cluster licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Enabled        perpetual
Security Contexts              : 10             perpetual
GTP/GPRS                       : Enabled        perpetual
AnyConnect Premium Peers          : 4              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled        perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 4              perpetual
Total UC Proxy Sessions        : 4              perpetual
Botnet Traffic Filter          : Enabled        33 days
Intercompany Media Engine      : Disabled       perpetual

This platform has an ASA 5520 VPN Plus license.

Running Permanent Activation Key: 0xyadayad1 0xyadayad1 0xyadayad1 0xyadayad1 0xyadayad1

Licensed features for this platform:
Maximum Physical Interfaces    : Unlimited      perpetual
Maximum VLANs                  : 150            perpetual
Inside Hosts                   : Unlimited      perpetual
Failover                       : Active/Active  perpetual
VPN-DES                        : Enabled        perpetual
VPN-3DES-AES                   : Disabled       perpetual
Security Contexts              : 2              perpetual
```

```
GTP/GPRS                       : Disabled       perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 750            perpetual
Total VPN Peers                   : 750            perpetual
Shared License                    : Disabled        perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions        : 2              perpetual
Total UC Proxy Sessions        : 2              perpetual
Botnet Traffic Filter          : Disabled       perpetual
Intercompany Media Engine      : Disabled       perpetual

The flash permanent activation key is the SAME as the running permanent key.
```

Example 5: Primary Unit Output for the ASA Services Module in a Failover Pair for show activation-key

The following is sample output from the **show activation-key** command for the primary failover unit that shows:

- The primary unit license (the combined permanent license and time-based licenses).
- The “Failover Cluster” license, which is the combined licenses from the primary and secondary units. This is the license that is actually running on the ASA. The values in this license that reflect the combination of the primary and secondary licenses are in bold.
- The primary unit installed time-based licenses (active and inactive).

```
ciscoasa# show activation-key

erial Number:  SAL144705BF
Running Permanent Activation Key: 0x4d1ed752 0xc8cfeb37 0xf4c38198 0x93c04c28 0x4a1c049a
Running Timebased Activation Key: 0xbc07bbd7 0xb15591e0 0xed68c013 0xd79374ff 0x44f87880

Licensed features for this platform:
Maximum Interfaces                : 1024           perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
DES                               : Enabled        perpetual
3DES-AES                          : Enabled        perpetual
Security Contexts                 : 25             perpetual
GTP/GPRS                          : Enabled        perpetual
Botnet Traffic Filter             : Enabled        330 days

This platform has an WS-SVC-ASA-SM1 No Payload Encryption license.


Failover cluster licensed features for this platform:
Maximum Interfaces                : 1024           perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
DES                               : Enabled        perpetual
3DES-AES                          : Enabled        perpetual
Security Contexts                 : 50 perpetual
GTP/GPRS                          : Enabled        perpetual
Botnet Traffic Filter             : Enabled        330 days
This platform has an WS-SVC-ASA-SM1 No Payload Encryption license.

The flash permanent activation key is the SAME as the running permanent key.
```

```
Active Timebased Activation Key:
0xbc07bbd7 0xb15591e0 0xed68c013 0xd79374ff 0x44f87880
Botnet Traffic Filter           : Enabled    330 days
```

Example 6: Secondary Unit Output for the ASA Services Module in a Failover Pair for show activation-key

The following is sample output from the **show activation-key** command for the secondary failover unit that shows:

- The secondary unit license (the combined permanent license and time-based licenses).
- The "Failover Cluster" license, which is the combined licenses from the primary and secondary units. This is the license that is actually running on the ASA. The values in this license that reflect the combination of the primary and secondary licenses are in bold.
- The secondary installed time-based licenses (active and inactive). This unit does not have any time-based licenses, so none display in this sample output.

```
ciscoasa# show activation-key detail

Serial Number:  SAD143502E3
Running Permanent Activation Key: 0xf404c46a 0xb8e5bd84 0x28c1b900 0x92eca09c 0x4e2a0683

Licensed features for this platform:
Maximum Interfaces                : 1024           perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
DES                               : Enabled        perpetual
3DES-AES                          : Enabled        perpetual
Security Contexts                 : 25             perpetual
GTP/GPRS                          : Disabled       perpetual
Botnet Traffic Filter             : Disabled       perpetual

This platform has an WS-SVC-ASA-SM1 No Payload Encryption license.


Failover cluster licensed features for this platform:
Maximum Interfaces                : 1024           perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
DES                               : Enabled        perpetual
3DES-AES                          : Enabled        perpetual
Security Contexts                 : 50 perpetual
GTP/GPRS                          : Enabled perpetual
Botnet Traffic Filter             : Enabled 330 days

This platform has an WS-SVC-ASA-SM1 No Payload Encryption license.

The flash permanent activation key is the SAME as the running permanent key.
```

Example 7: Output in a Cluster for show activation-key

```
ciscoasa# show activation-key
 Serial Number: JMX1504L2TD
 Running Permanent Activation Key: 0x4a3eea7b 0x54b9f61a 0x4143a90c 0xe5849088 0x4412d4a9

 Licensed features for this platform:
```

```
Maximum Physical Interfaces : Unlimited perpetual
Maximum VLANs : 100 perpetual
Inside Hosts : Unlimited perpetual
Failover : Active/Active perpetual
Encryption-DES : Enabled perpetual
Encryption-3DES-AES : Enabled perpetual
Security Contexts : 2 perpetual
GTP/GPRS : Disabled perpetual
AnyConnect Premium Peers : 2 perpetual
AnyConnect Essentials : Disabled perpetual
Other VPN Peers : 250 perpetual
Total VPN Peers : 250 perpetual
Shared License : Disabled perpetual
AnyConnect for Mobile : Disabled perpetual
AnyConnect for Cisco VPN Phone : Disabled perpetual
Advanced Endpoint Assessment : Disabled perpetual
UC Phone Proxy Sessions : 2 perpetual
Total UC Proxy Sessions : 2 perpetual
Botnet Traffic Filter : Disabled perpetual
Intercompany Media Engine : Disabled perpetual
Cluster : Enabled perpetual

This platform has an ASA 5585-X base license.

Failover cluster licensed features for this platform:
Maximum Physical Interfaces : Unlimited perpetual
Maximum VLANs : 100 perpetual
Inside Hosts : Unlimited perpetual
Failover : Active/Active perpetual
Encryption-DES : Enabled perpetual
Encryption-3DES-AES : Enabled perpetual
Security Contexts : 4 perpetual
GTP/GPRS : Disabled perpetual
AnyConnect Premium Peers : 4 perpetual
AnyConnect Essentials : Disabled perpetual
Other VPN Peers : 250 perpetual
Total VPN Peers : 250 perpetual
Shared License : Disabled perpetual
AnyConnect for Mobile : Disabled perpetual
AnyConnect for Cisco VPN Phone : Disabled perpetual
Advanced Endpoint Assessment : Disabled perpetual
UC Phone Proxy Sessions : 4 perpetual
Total UC Proxy Sessions : 4 perpetual
Botnet Traffic Filter : Disabled perpetual
Intercompany Media Engine : Disabled perpetual
Cluster : Enabled perpetual

This platform has an ASA 5585-X base license.

The flash permanent activation key is the SAME as the running permanent key.
```

Monitoring the Shared License

To monitor the shared license, enter one of the following commands.

- **show shared license** [**detail** | **client** [*hostname*] | **backup**]

  Shows shared license statistics. Optional keywords are available only for the licensing server: the **detail** keyword shows statistics per participant. To limit the display to one participant, use the **client** keyword. The **backup** keyword shows information about the backup server.

  To clear the shared license statistics, enter the **clear shared license** command.

The following is sample output from the **show shared license** command on the license participant:

```
ciscoasa>  show shared license
Primary License Server : 10.3.32.20
  Version              : 1
  Status               : Inactive

Shared license utilization:
  SSLVPN:
    Total for network :     5000
    Available         :     5000
    Utilized          :        0
  This device:
    Platform limit    :      250
    Current usage     :        0
    High usage        :        0
  Messages Tx/Rx/Error:
    Registration    : 0 / 0 / 0
    Get             : 0 / 0 / 0
    Release         : 0 / 0 / 0
    Transfer        : 0 / 0 / 0
```

The following is sample output from the **show shared license detail** command on the license server:

```
ciscoasa>  show shared license detail
Backup License Server Info:

Device ID           : ABCD
Address             : 10.1.1.2
Registered          : NO
HA peer ID          : EFGH
Registered          : NO
  Messages Tx/Rx/Error:
    Hello           : 0 / 0 / 0
    Sync            : 0 / 0 / 0
    Update          : 0 / 0 / 0

Shared license utilization:
  SSLVPN:
    Total for network :      500
    Available         :      500
    Utilized          :        0
  This device:
    Platform limit    :      250
    Current usage     :        0
    High usage        :        0
  Messages Tx/Rx/Error:
    Registration    : 0 / 0 / 0
    Get             : 0 / 0 / 0
    Release         : 0 / 0 / 0
    Transfer        : 0 / 0 / 0


Client Info:

  Hostname          : 5540-A
  Device ID         : XXXXXXXXXXX
  SSLVPN:
    Current usage   : 0
    High            : 0
  Messages Tx/Rx/Error:
    Registration    : 1 / 1 / 0
    Get             : 0 / 0 / 0
    Release         : 0 / 0 / 0
    Transfer        : 0 / 0 / 0
...
```

- **show activation-key**

  Shows the licenses installed on the ASA. The **show version** command also shows license information.

- **show vpn-sessiondb**

  Shows license information about VPN sessions.

History for PAK Licenses

Feature Name	Platform Releases	Description
Increased Connections and VLANs	7.0(5)	Increased the following limits: • ASA5510 Base license connections from 32000 to 5000; VLANs from 0 to 10. • ASA5510 Security Plus license connections from 64000 to 130000; VLANs from 10 to 25. • ASA5520 connections from 130000 to 280000; VLANs from 25 to 100. • ASA5540 connections from 280000 to 400000; VLANs from 100 to 200.
SSL VPN Licenses	7.1(1)	SSL VPN licenses were introduced.
Increased SSL VPN Licenses	7.2(1)	A 5000-user SSL VPN license was introduced for the ASA 5550 and above.
Increased interfaces for the Base license on the ASA 5510	7.2(2)	For the Base license on the ASA 5510, the maximum number of interfaces was increased from 3 plus a management interface to unlimited interfaces.

Feature Name	Platform Releases	Description
Increased VLANs	7.2(2)	The maximum number of VLANs for the Security Plus license on the ASA 5505 was increased from 5 (3 fully functional; 1 failover; one restricted to a backup interface) to 20 fully functional interfaces. In addition, the number of trunk ports was increased from 1 to 8. Now there are 20 fully functional interfaces, you do not need to use the backup interface command to cripple a backup ISP interface; you can use a fully functional interface for it. The backup interface command is still useful for an Easy VPN configuration. VLAN limits were also increased for the ASA 5510 (from 10 to 50 for the Base license, and from 25 to 100 for the Security Plus license), the ASA 5520 (from 100 to 150), the ASA 5550 (from 200 to 250).
Gigabit Ethernet Support for the ASA 5510 Security Plus License	7.2(3)	The ASA 5510 now supports Gigabit Ethernet (1000 Mbps) for the Ethernet 0/0 and 0/1 ports with the Security Plus license. In the Base license, they continue to be used as Fast Ethernet (100 Mbps) ports. Ethernet 0/2, 0/3, and 0/4 remain as Fast Ethernet ports for both licenses. **Note** The interface names remain Ethernet 0/0 and Ethernet 0/1. Use the **speed** command to change the speed on the interface and use the **show interface** command to see what speed is currently configured for each interface.

Feature Name	Platform Releases	Description
Advanced Endpoint Assessment License	8.0(2)	The Advanced Endpoint Assessment license was introduced. As a condition for the completion of a Cisco AnyConnect or clientless SSL VPN connections, the remote computer scans for a greatly expanded collection of antivirus and antispyware applications, firewalls, operating systems, and associated updates. It also scans for any registry entries, filenames, and process names that you specify. It sends the scan results to the ASA. The ASA uses both the user login credentials and the computer scan results to assign a Dynamic Access Policy (DAP). With an Advanced Endpoint Assessment License, you can enhance Host Scan by configuring an attempt to update noncompliant computers to meet version requirements. Cisco can provide timely updates to the list of applications and versions that Host Scan supports in a package that is separate from Cisco Secure Desktop.
VPN Load Balancing for the ASA 5510	8.0(2)	VPN load balancing is now supported on the ASA 5510 Security Plus license.
AnyConnect for Mobile License	8.0(3)	The AnyConnect for Mobile license was introduced. It lets Windows mobile devices connect to the ASA using the AnyConnect client.
Time-based Licenses	8.0(4)/8.1(2)	Support for time-based licenses was introduced.
Increased VLANs for the ASA 5580	8.1(2)	The number of VLANs supported on the ASA 5580 are increased from 100 to 250.
Unified Communications Proxy Sessions license	8.0(4)	The UC Proxy sessions license was introduced. Phone Proxy, Presence Federation Proxy, and Encrypted Voice Inspection applications use TLS proxy sessions for their connections. Each TLS proxy session is counted against the UC license limit. All of these applications are licensed under the UC Proxy umbrella, and can be mixed and matched. This feature is not available in Version 8.1.

Feature Name	Platform Releases	Description
Botnet Traffic Filter License	8.2(1)	The Botnet Traffic Filter license was introduced. The Botnet Traffic Filter protects against malware network activity by tracking connections to known bad domains and IP addresses.
AnyConnect Essentials License	8.2(1)	The AnyConnect Essentials License was introduced. This license enables AnyConnect VPN client access to the ASA. This license does not support browser-based SSL VPN access or Cisco Secure Desktop. For these features, activate an AnyConnect Premium license instead of the AnyConnect Essentials license. **Note** With the AnyConnect Essentials license, VPN users can use a Web browser to log in, and download and start (WebLaunch) the AnyConnect client. The AnyConnect client software offers the same set of client features, whether it is enabled by this license or an AnyConnect Premium license. The AnyConnect Essentials license cannot be active at the same time as the following licenses on a given ASA: AnyConnect Premium license (all types) or the Advanced Endpoint Assessment license. You can, however, run AnyConnect Essentials and AnyConnect Premium licenses on different ASAs in the same network. By default, the ASA uses the AnyConnect Essentials license, but you can disable it to use other licenses by using the **webvpn**, and then the **no anyconnect-essentials** command.
SSL VPN license changed to AnyConnect Premium SSL VPN Edition license	8.2(1)	The SSL VPN license name was changed to the AnyConnect Premium SSL VPN Edition license.
Shared Licenses for SSL VPN	8.2(1)	Shared licenses for SSL VPN were introduced. Multiple ASAs can share a pool of SSL VPN sessions on an as-needed basis.

Feature Name	Platform Releases	Description
Mobility Proxy application no longer requires Unified Communications Proxy license	8.2(2)	The Mobility Proxy no longer requires the UC Proxy license.
10 GE I/O license for the ASA 5585-X with SSP-20	8.2(3)	We introduced the 10 GE I/O license for the ASA 5585-X with SSP-20 to enable 10-Gigabit Ethernet speeds for the fiber ports. The SSP-60 supports 10-Gigabit Ethernet speeds by default. **Note** The ASA 5585-X is not supported in 8.3(x).
10 GE I/O license for the ASA 5585-X with SSP-10	8.2(4)	We introduced the 10 GE I/O license for the ASA 5585-X with SSP-10 to enable 10-Gigabit Ethernet speeds for the fiber ports. The SSP-40 supports 10-Gigabit Ethernet speeds by default. **Note** The ASA 5585-X is not supported in 8.3(x).
Non-identical failover licenses	8.3(1)	Failover licenses no longer need to be identical on each unit. The license used for both units is the combined license from the primary and secondary units. We modified the following commands: **show activation-key** and **show version**.
Stackable time-based licenses	8.3(1)	Time-based licenses are now stackable. In many cases, you might need to renew your time-based license and have a seamless transition from the old license to the new one. For features that are only available with a time-based license, it is especially important that the license not expire before you can apply the new license. The ASA allows you to *stack* time-based licenses so that you do not have to worry about the license expiring or about losing time on your licenses because you installed the new one early.
Intercompany Media Engine License	8.3(1)	The IME license was introduced.

Feature Name	Platform Releases	Description
Multiple time-based licenses active at the same time	8.3(1)	You can now install multiple time-based licenses, and have one license per feature active at a time. We modified the following commands: **show activation-key** and **show version**.
Discrete activation and deactivation of time-based licenses.	8.3(1)	You can now activate or deactivate time-based licenses using a command. We modified the following commands: **activation-key** [**activate** | **deactivate**].
AnyConnect Premium SSL VPN Edition license changed to AnyConnect Premium SSL VPN license	8.3(1)	The AnyConnect Premium SSL VPN Edition license name was changed to the AnyConnect Premium SSL VPN license.
No Payload Encryption image for export	8.3(2)	If you install the No Payload Encryption software on the ASA 5505 through 5550, then you disable Unified Communications, strong encryption VPN, and strong encryption management protocols. **Note** This special image is only supported in 8.3(x); for No Payload Encryption support in 8.4(1) and later, you need to purchase a special hardware version of the ASA.
Increased contexts for the ASA 5550, 5580, and 5585-X	8.4(1)	For the ASA 5550 and ASA 5585-X with SSP-10, the maximum contexts was increased from 50 to 100. For the ASA 5580 and 5585-X with SSP-20 and higher, the maximum was increased from 50 to 250.
Increased VLANs for the ASA 5580 and 5585-X	8.4(1)	For the ASA 5580 and 5585-X, the maximum VLANs was increased from 250 to 1024.

Feature Name	Platform Releases	Description
Increased connections for the ASA 5580 and 5585-X	8.4(1)	We increased the firewall connection limits: • ASA 5580-20—1,000,000 to 2,000,000. • ASA 5580-40—2,000,000 to 4,000,000. • ASA 5585-X with SSP-10: 750,000 to 1,000,000. • ASA 5585-X with SSP-20: 1,000,000 to 2,000,000. • ASA 5585-X with SSP-40: 2,000,000 to 4,000,000. • ASA 5585-X with SSP-60: 2,000,000 to 10,000,000.
AnyConnect Premium SSL VPN license changed to AnyConnect Premium license	8.4(1)	The AnyConnect Premium SSL VPN license name was changed to the AnyConnect Premium license. The license information display was changed from “SSL VPN Peers” to “AnyConnect Premium Peers.”
Increased AnyConnect VPN sessions for the ASA 5580	8.4(1)	The AnyConnect VPN session limit was increased from 5,000 to 10,000.
Increased Other VPN sessions for the ASA 5580	8.4(1)	The other VPN session limit was increased from 5,000 to 10,000.
IPsec remote access VPN using IKEv2	8.4(1)	IPsec remote access VPN using IKEv2 was added to the AnyConnect Essentials and AnyConnect Premium licenses. **Note** The following limitation exists in our support for IKEv2 on the ASA: We currently do not support duplicate security associations. IKEv2 site-to-site sessions were added to the Other VPN license (formerly IPsec VPN). The Other VPN license is included in the Base license.

Feature Name	Platform Releases	Description
No Payload Encryption hardware for export	8.4(1)	For models available with No Payload Encryption (for example, the ASA 5585-X), the ASA software disables Unified Communications and VPN features, making the ASA available for export to certain countries.
Dual SSPs for SSP-20 and SSP-40	8.4(2)	For SSP-40 and SSP-60, you can use two SSPs of the same level in the same chassis. Mixed-level SSPs are not supported (for example, an SSP-40 with an SSP-60 is not supported). Each SSP acts as an independent device, with separate configurations and management. You can use the two SSPs as a failover pair if desired. When using two SSPs in the chassis, VPN is not supported; note, however, that VPN has not been disabled.
IPS Module license for the ASA 5512-X through ASA 5555-X	8.6(1)	The IPS SSP software module on the ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, and ASA 5555-X requires the IPS module license.
Clustering license for the ASA 5580 and ASA 5585-X.	9.0(1)	A clustering license was added for the ASA 5580 and ASA 5585-X.
Support for VPN on the ASASM	9.0(1)	The ASASM now supports all VPN features.
Unified communications support on the ASASM	9.0(1)	The ASASM now supports all Unified Communications features.
ASA 5585-X Dual SSP support for the SSP-10 and SSP-20 (in addition to the SSP-40 and SSP-60); VPN support for Dual SSPs	9.0(1)	The ASA 5585-X now supports dual SSPs using all SSP models (you can use two SSPs of the same level in the same chassis). VPN is now supported when using dual SSPs.
ASA 5500-X support for clustering	9.1(4)	The ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, and ASA 5555-X now support 2-unit clusters. Clustering for 2 units is enabled by default in the base license; for the ASA 5512-X, you need the Security Plus license.
Support for 16 cluster members for the ASA 5585-X	9.2(1)	The ASA 5585-X now supports 16-unit clusters.

Feature Name	Platform Releases	Description
ASAv4 and ASAv30 Standard and Premium model licenses introduced	9.2(1)	The ASAv was introduced with a simple licensing scheme: ASAv4 and ASAv30 permanent licenses in Standard or Premium levels. No add-on licenses are available.

CHAPTER 4

Licenses: Smart Software Licensing (ASAv, ASA on Firepower)

Cisco Smart Software Licensing lets you purchase and manage a pool of licenses centrally. Unlike product authorization key (PAK) licenses, smart licenses are not tied to a specific serial number. You can easily deploy or retire ASAs without having to manage each unit's license key. Smart Software Licensing also lets you see your license usage and needs at a glance.

Note Smart Software Licensing is only supported on the ASAv and ASA Firepower chassis. Other models use PAK licenses. See About PAK Licenses, on page 49.

For more information about Smart Licensing features and behaviors per platform, see Smart Enabled Product Families.

About Smart Software Licensing

This section describes how Smart Software Licensing works.

Smart Software Licensing for the ASA on the Firepower 9300 Chassis

For the ASA on the Firepower 9300 chassis, Smart Software Licensing configuration is split between the Firepower 9300 chassis supervisor and the ASA.

- Firepower 9300 chassis—Configure all Smart Software Licensing infrastructure on the chassis, including parameters for communicating with the License Authority. The Firepower 9300 chassis itself does not require any licenses to operate.
- ASA Application—Configure all license entitlements in the ASA.

Smart Software Manager and Accounts

When you purchase 1 or more licenses for the device, you manage them in the Cisco Smart Software Manager:

https://software.cisco.com/#module/SmartLicensing

The Smart Software Manager lets you create a master account for your organization.

Note If you do not yet have an account, click the link to set up a new account. The Smart Software Manager lets you create a master account for your organization.

By default, your licenses are assigned to the *Default Virtual Account* under your master account. As the account administrator, you can optionally create additional virtual accounts; for example, you can create accounts for regions, departments, or subsidiaries. Multiple virtual accounts let you more easily manage large numbers of licenses and devices.

Licenses and Devices Managed per Virtual Account

Licenses and devices are managed per virtual account: only that virtual account's devices can use the licenses assigned to the account. If you need additional licenses, you can transfer an unused license from another virtual account. You can also transfer devices between virtual accounts.

For the ASA on the Firepower 9300 chassis—Only the chassis registers as a device, while the ASA applications in the chassis request their own licenses. For example, for a Firepower 9300 chassis with 3 security modules, the chassis counts as one device, but the modules use 3 separate licenses.

Evaluation License

ASAv

The ASAv does not support an evaluation mode. Before the ASAv registers with the Licensing Authority, it operates in a severely rate-limited state.

Firepower 9300 Chassis

The Firepower 9300 chassis supports two types of evaluation license:

- Chassis-level evaluation mode—Before the Firepower 9300 chassis registers with the Licensing Authority, it operates for 90 days (total usage) in evaluation mode. The ASA cannot request specific entitlements in this mode; only default entitlements are enabled. When this period ends, the Firepower 9300 chassis becomes out-of-compliance.
- Entitlement-based evaluation mode—After the Firepower 9300 chassis registers with the Licensing Authority, you can obtain time-based evaluation licenses that can be assigned to the ASA. In the ASA,

you request entitlements as usual. When the time-based license expires, you need to either renew the time-based license or obtain a permanent license.

Note You cannot receive an evaluation license for Strong Encryption (3DES/AES); you must register with the License Authority and obtain a permanent license.

About Licenses by Type

The following sections include additional information about licenses by type.

AnyConnect Plus, AnyConnect Apex, And VPN Only Licenses

The AnyConnect Plus, AnyConnect Apex, or VPN Only license is a multi-use license that you can apply to multiple ASAs, all of which share a user pool as specified by the license. Devices that use Smart Licensing do not require any AnyConnect license to be physically applied to the actual platform. The same licenses must still be purchased, and you must still link the Contract number to your Cisco.com ID for SW Center access and technical support. For more information, see:

- Cisco AnyConnect Ordering Guide
- AnyConnect Licensing Frequently Asked Questions (FAQ)

Other VPN License

Other VPN sessions include the following VPN types:

- IPsec remote access VPN using IKEv1
- IPsec site-to-site VPN using IKEv1
- IPsec site-to-site VPN using IKEv2

This license is included in the Base license.

Total VPN Sessions Combined, All Types

- Although the maximum VPN sessions add up to more than the maximum VPN AnyConnect and Other VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the ASA, so be sure to size your network appropriately.
- If you start a clientless SSL VPN session and then start an AnyConnect client session from the portal, 1 session is used in total. However, if you start the AnyConnect client first (from a standalone client, for example) and then log into the clientless SSL VPN portal, then 2 sessions are used.

Encryption License

Strong Encryption: ASAv

Strong Encryption (3DES/AES) is available for management connections before you connect to the License Authority, so you can launch ASDM and connect to the License Authority. For through-the-box traffic, throughput is severely limited until you connect to the License Authority and obtain the Strong Encryption license.

If the ASAv becomes out-of-compliance later, then the ASAv reverts to the rate-limited state.

Strong Encryption: Firepower 9300 Chassis

You must manually request the Strong Encryption license in the ASA configuration using the CLI because ASDM requires 3DES. If the ASA becomes out-of-compliance, neither management traffic nor through-traffic requiring this license will be allowed.

DES: All Models

The DES license cannot be disabled. If you have the 3DES license installed, DES is still available. To prevent the use of DES when you want to only use strong encryption, be sure to configure any relevant commands to use only strong encryption.

Total UC Proxy Sessions

Each TLS proxy session for Encrypted Voice Inspection is counted against the TLS license limit.

Other applications that use TLS proxy sessions do not count toward the TLS limit, for example, Mobility Advantage Proxy (which does not require a license).

Some applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command or in ASDM, using the **Configuration** > **Firewall** > **Unified Communications** > **TLS Proxy** pane. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a TLS proxy license that is higher than the default TLS proxy limit, the ASA automatically sets the TLS proxy limit to match the license. The TLS proxy limit takes precedence over the license limit; if you set the TLS proxy limit to be less than the license, then you cannot use all of the sessions in your license.

Note For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the license limit, then you see an error message to use the **tls-proxy maximum-sessions** command to raise the limit again (in ASDM, use the **TLS Proxy** pane). If you use failover and enter the **write standby** command or in ASDM, use **File > Save Running Configuration to Standby Unit** on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:

- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is no limit.

Note Only calls that require encryption/decryption for media are counted toward the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count toward the limit.

VLANs, Maximum

For an interface to count against the VLAN limit, you must assign a VLAN to it. For example:

```
interface gigabitethernet 0/0.100
vlan 100
```

Botnet Traffic Filter License

Requires a Strong Encryption (3DES/AES) License to download the dynamic database.

Failover or ASA Cluster Licenses

Failover Licenses for the ASAv

The standby unit requires the same model license as the primary unit.

Failover Licenses for the ASA on the Firepower 9300 Chassis

Each Firepower 9300 chassis must be registered with the License Authority or satellite server. There is no extra cost for the secondary unit. For permanent license reservation, you must purchase separate licenses for each chassis.

Each ASA must have the same encryption license. For regular Smart Software Manager users, the Strong Encryption license is automatically enabled for qualified customers when you apply the registration token on the Firepower 9300 chassis. For older Cisco Smart Software Manager satellite deployments, see below.

In the ASA licensing configuration, other licenses do not need to match on each failover unit, and you can configure licensing separately on each unit. Each unit requests its own licenses from the server. The licenses requested by both units are aggregated into a single failover license that is shared by the failover pair, and this aggregated licenese is cached on the standby unit to be used if it becomes the active unit in the future. Typically, you only need to configure licenses on the primary unit.

Each license type is managed as follows:

- Standard—Each unit includes the Standard license by default, so for a failover pair, 2 Standard licenses are requested from the server.
- Context—Each unit can request its own Context license. However, the Standard license includes 10 contexts by default and is present on both units. The value from each unit's Standard license plus the value of any optional Context licenses on both units are combined up to the platform limit. For example:
  - The Standard license includes 10 contexts; for 2 units, these licenses add up to 20 contexts.You configure a 250-Context license on the primary unit in an Active/Standby pair. Therefore, the aggregated failover license includes 270 contexts. However, because the platform limit for one unit is 250, the combined license allows a maximum of 250 contexts only. In this case, you should only configure the primary Context license to be 230 contexts.

  - The Standard license includes 10 contexts; for 2 units, these licenses add up to 20 contexts. You configure a 10-Context license on the primary unit in an Active/Active pair, and a 10-Context license on the secondary unit. Therefore, the aggregated failover license includes 40 contexts. One unit can use 22 contexts and the other unit can use 18 contexts, for example, for a total of 40. Because the platform limit for one unit is 250, the combined license allows a maximum of 250 contexts; the 40 contexts are within the limit.

- Carrier—Only one unit needs to request this license, and both units can use it.

- Strong Encryption (3DES) (for a pre-2.3.0 Cisco Smart Software Manager satellite deployment only)—Each unit must request its own license from the server; unlike the other license configurations, this configuration is replicated to the standby unit. For Smart Software Manager satellite deployments, to use ASDM and other strong encryption features, after you deploy the cluster you must enable the Strong Encryption (3DES) license on the primary unit using the ASA CLI. The Strong Encryption (3DES) license is not available with any type of evaluation license.

ASA Cluster Licenses for the ASA on the Firepower 9300 Chassis

The clustering feature itself does not require any licenses. To use Strong Encryption and other optional licenses, you can only request licenses on the control unit; the licenses are aggregated with the data units. If you have licenses on multiple units, they combine into a single running ASA cluster license. License configuration completed on the control unit is not replicated to the data units. You can only configure separate license entitlements on data units if you disable clustering, configure the licensing, and then re-enable clustering.

Note To use ASDM and other strong encryption features, after you deploy the cluster you must enable the Strong Encryption (3DES) license on the control unit using the ASA CLI. This license is inherited by the data units; you do not need to configure this license separately on each unit. The Strong Encryption (3DES) license is not available with any type of evaluation license.

Note If the control unit fails, and does not rejoin within 30 days (the licensing grace period), then the inherited licenses disappear. You must then manually configure the missing licenses on the new control unit.

Prerequisites for Smart Software Licensing

Regular and Satellite Smart License Prerequisites

ASAv

- Ensure internet access, or HTTP proxy access from the device.

- Configure a DNS server so the device can resolve the name of the License Authority.

- Set the clock for the device.

- Create a master account on the Cisco Smart Software Manager:

https://software.cisco.com/#module/SmartLicensing

If you do not yet have an account, click the link to set up a new account. The Smart Software Manager lets you create a master account for your organization.

Firepower 4100/9300

Configure the Smart Software Licensing infrastructure on the Firepower 9300 chassis before you configure the ASA licensing entitlements.

Permanent License Reservation Prerequisites

- Create a master account on the Cisco Smart Software Manager:

  https://software.cisco.com/#module/SmartLicensing

  If you do not yet have an account, click the link to set up a new account. The Smart Software Manager lets you create a master account for your organization. Even though the ASA does need internet connectivity to the Smart Licensing server for permanent license reservation, the Smart Software Manager is used to manage your permanent licenses.

- Obtain support for permanent license reservation from the licensing team. You must provide a justification for using permanent license reservation. If your account is not approved, then you cannot purchase and apply permanent licenses.

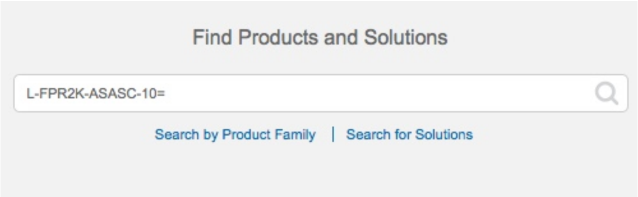
- Purchase special permanent licenses (see License PIDs, on page 115). If you do not have the correct license in your account, then when you try to reserve a license on the ASA, you will see an error message similar to: "The licenses cannot be reserved because the Virtual Account does not contain a sufficient surplus of the following perpetual licenses: 1 - Firepower 4100 ASA PERM UNIV(perpetual)."

- The permanent license includes all available features, including the Strong Encryption (3DES/AES) license if your account qualifies. AnyConnect client capabilities are also enabled to the platform maximum, contingent on your purchase of an AnyConnect license that enables the right to use AnyConnect (see AnyConnect Plus, AnyConnect Apex, And VPN Only Licenses, on page 111).

License PIDs

When you bought your device from Cisco or a reseller, your licenses should have been linked to your Smart Software License account. However, if you need to add licenses yourself, use the **Find Products and Solutions** search field on the Cisco Commerce Workspace. Search for the following license Product IDs (PIDs).

Figure 7: License Search



ASAv PIDs

ASAv PIDs:

- ASAv5—L-ASAV5S-K9=
- ASAv10—L-ASAV10S-K9=
- ASAv30—L-ASAV30S-K9=
- ASAv50—L-ASAV50S-K9=

Firepower 9300 PIDs

Firepower 9300 PIDs:

- Standard license—L-F9K-ASA=. The Standard license is free, but you still need to add it to your Smart Software Licensing account.
- 10 context license—L-F9K-ASA-SC-10=. Context licenses are additive; buy multiple licenses to meet your needs.
- Carrier (Diameter, GTP/GPRS, SCTP)—L-F9K-ASA-CAR=
- Strong Encryption (3DES/AES) license—L-F9K-ASA-ENCR-K9=. This license is free. Although this license is not generally rquired (for example, ASAs that use older Satellite Server versions (pre-2.3.0) require this license), you should still add it to your account for tracking purposes.

Guidelines for Smart Software Licensing

- Only Smart Software Licensing is supported. For older software on the ASAv, if you upgrade an existing PAK-licensed ASAv, then the previously installed activation key will be ignored, but retained on the device. If you downgrade the ASAv, the activation key will be reinstated.
- (Firepower 9300 ASA security module) To use ASDM and other strong encryption features such as VPN, after you deploy the ASA you must enable the Strong Encryption (3DES) license on the control unit using the ASA CLI. For clustering, configure the license on the control unit. This license is inherited by the data units; you do not need to configure this license separately on each unit.
- Because the Cisco Transport Gateway uses a certificate with a non-compliant country code, you cannot use HTTPS when using the ASA in conjunction with that product. You must use HTTP with Cisco Transport Gateway.

Defaults for Smart Software Licensing

ASAv

- The ASAv default configuration includes a Smart Call Home profile called “License” that specifies the URL for the Licensing Authority.

```
call-home
  profile License
    destination address http
https://tools.cisco.com/its/service/oddce/services/DDCEService
```

- When you deploy the ASAv, you set the feature tier and throughput level. Only the standard level is available at this time.

```
license smart
  feature tier standard
  throughput level {100M | 1G | 2G}
```

- Also during deployment, you can optionally configure an HTTP proxy.

```
call-home
  http-proxy ip_address port port
```

ASA on the Firepower 9300 Chassis

There is no default configuration. You must manually enable the standard license tier and other optional licenses.

ASAv: Configure Smart Software Licensing

This section describes how to configure Smart Software Licensing for the ASAv.

Procedure

ASAv: Configure Smart Software Licensing, on page 117.

ASAv: Configure Smart Software Licensing

When you deploy the ASAv, you can pre-configure the device and include a registration token so it registers with the License Authority and enables Smart Software Licensing. If you need to change your HTTP proxy server, license entitlement, or register the ASAv (for example, if you did not include the ID token in the Day0 configuration), perform this task.

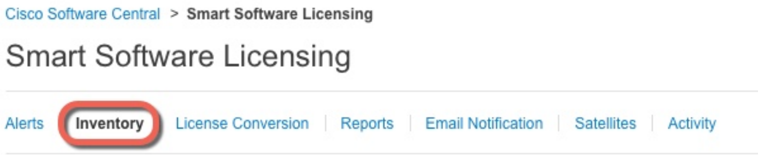
Note You may have pre-configured the HTTP proxy and license entitlements when you deployed your ASAv. You may also have included the registration token with your Day0 configuration when you deployed the ASAv; if so, you do not need to re-register using this procedure.

Procedure

Step 1 In the Smart Software Manager (Cisco Smart Software Manager), request and copy a registration token for the virtual account to which you want to add this device.

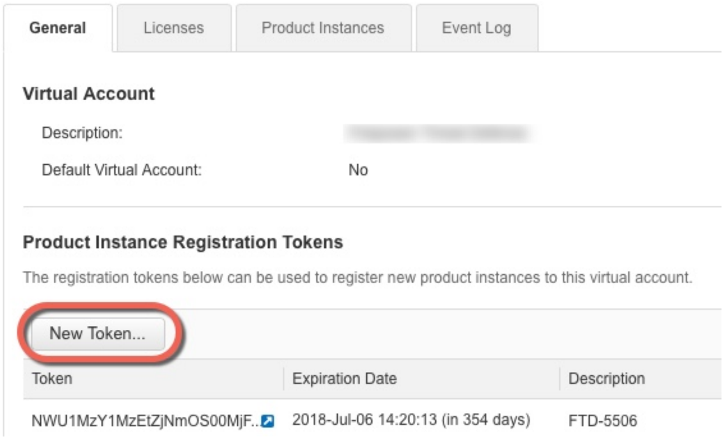
a) Click **Inventory**.

Figure 8: Inventory



b) On the **General** tab, click **New Token**.

Figure 9: New Token



Virtual Account

Description: [redacted]

Default Virtual Account: No

Product Instance Registration Tokens

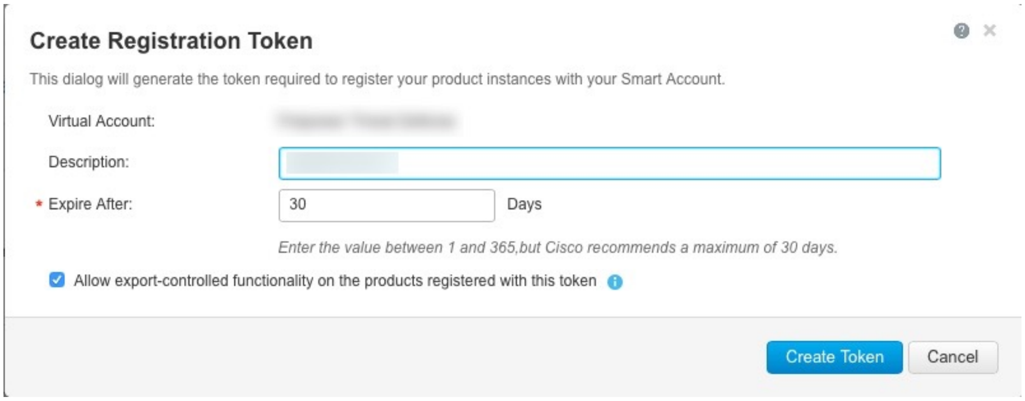
The registration tokens below can be used to register new product instances to this virtual account.

Token	Expiration Date	Description
NWU1MzY1MzEtZjNmOS00MjF...	2018-Jul-06 14:20:13 (in 354 days)	FTD-5506

c) On the **Create Registration Token** dialog box enter the following settings, and then click **Create Token**:

- **Description**
- **Expire After**—Cisco recommends 30 days.
- **Allow export-controlled functionaility on the products registered with this token**—Enables the export-compliance flag.

Figure 10: Create Registration Token



This dialog will generate the token required to register your product instances with your Smart Account.

Virtual Account: [redacted]

Description: [redacted]

* Expire After: 30 Days

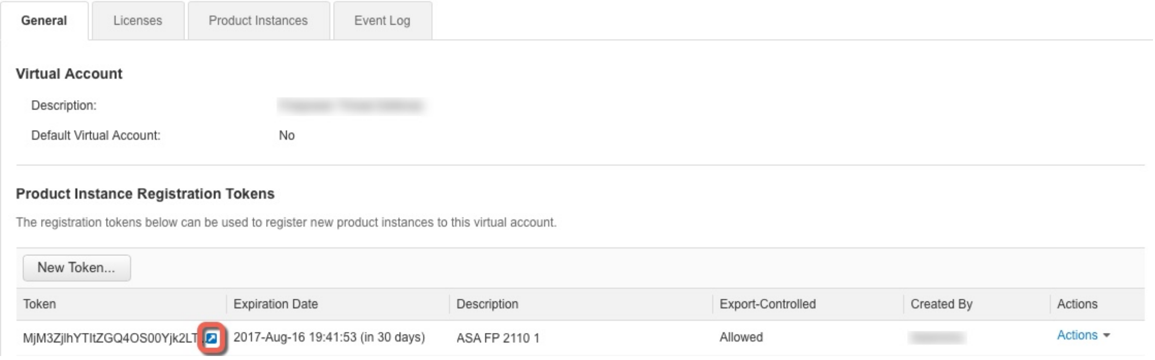
Enter the value between 1 and 365,but Cisco recommends a maximum of 30 days.

☑ Allow export-controlled functionality on the products registered with this token

The token is added to your inventory.

d) Click the arrow icon to the right of the token to open the **Token** dialog box so you can copy the token ID to your clipboard. Keep this token ready for later in the procedure when you need to register the ASA.

Figure 11: View Token



Virtual Account

Description: [redacted]

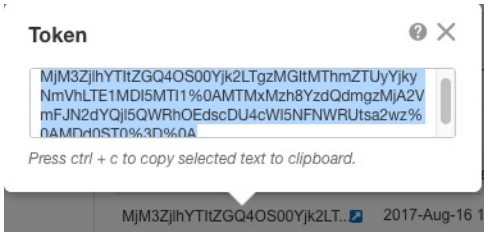
Default Virtual Account: No

Product Instance Registration Tokens

The registration tokens below can be used to register new product instances to this virtual account.

Token	Expiration Date	Description	Export-Controlled	Created By	Actions
MjM3ZjlhYTItZGQ4OS00Yjk2LT...	2017-Aug-16 19:41:53 (in 30 days)	ASA FP 2110 1	Allowed	[redacted]	Actions ▾

Figure 12: Copy Token



Token

```
MjM3ZjlhYTItZGQ4OS00Yjk2LTgzMGItMThmZTUyYjky
NmVhLTE1MDI5MTI1%0AMTMxMzh8YzdQdmgzMjA2V
mFJN2dYQjI5QWRhOEdscDU4cWI5NFNWRUtsa2wz%
0AMDd0ST0%3D%0A
```

Press ctrl + c to copy selected text to clipboard.

MjM3ZjlhYTItZGQ4OS00Yjk2LT.. 2017-Aug-16 1

Step 2 (Optional) On the ASAv, specify the HTTP Proxy URL:

call-home

http-proxy *ip_address* **port** *port*

If your network uses an HTTP proxy for internet access, you must configure the proxy address for Smart Software Licensing. This proxy is also used for Smart Call Home in general.

Example:

```
ciscoasa(config)# call-home
ciscoasa(cfg-call-home)# http-proxy 10.1.1.1 port 443
```

Step 3 Configure the license entitlements.

a) Enter license smart configuration mode:

license smart

Example:

```
ciscoasa(config)# license smart
ciscoasa(config-smart-lic)#
```

b) Set the feature tier:

feature tier standard

Only the standard tier is available.

c) Set the throughput level:

throughput level {**100M** | **1G** | **2G**}

Example:

```
ciscoasa(config-smart-lic)# throughput level 2G
```

a) Exit license smart mode to apply your changes:

exit

Your changes do not take effect until you exit the license smart configuration mode, either by explicitly exiting the mode (**exit** or **end**) or by entering any command that takes you to a different mode.

Example:

```
ciscoasa(config-smart-lic)# exit
ciscoasa(config)#
```

Step 4 Register the ASAv with the License Authority.

When you register the ASAv, the License Authority issues an ID certificate for communication between the ASAv and the License Authority. It also assigns the ASAv to the appropriate virtual account. Normally, this procedure is a one-time instance. However, you might need to later re-register the ASAv if the ID certificate expires because of a communication problem, for example.

a) Enter the registration token on the ASAv:

license smart register idtoken *id_token* [**force**]

Example:

Use the **force** keyword to register an ASAv that is already registered, but that might be out of sync with the License Authority. For example, use **force** if the ASAv was accidentally removed from the Smart Software Manager.

The ASAv attempts to register with the License Authority and request authorization for the configured license entitlements.

Example:

```
ciscoasa# license smart register idtoken YjE3Njc5MzYtMGQzMi00OTA4
LWJhODItNzBhMGQ5NGRlYjUxLTE0MTQ5NDAy%0AODQzNzl8NXk2bzV3SDE0ZkgwQk
dYRmZ1NTNCNGlvRnBHUFpjcm02WTB4TU4w%0Ac2NnMD0%3D%0A
```

(Optional) Deregister the ASAv

Deregistering the ASAv removes the ASAv from your account. All license entitlements and certificates on the ASAv are removed. You might want to deregister to free up a license for a new ASAv. Alternatively, you can remove the ASAv from the Smart Software Manager.

Procedure

Deregister the ASAv:

license smart deregister

The ASAv then reloads.

(Optional) Renew the ASAv ID Certificate or License Entitlement

By default, the ID certificate is automatically renewed every 6 months, and the license entitlement is renewed every 30 days. You might want to manually renew the registration for either of these items if you have a limited window for Internet access, or if you make any licensing changes in the Smart Software Manager, for example.

Procedure

Step 1 Renew the ID certificate:

license smart renew id

Step 2 Renew the license entitlement:

license smart renew auth

Firepower 4100/9300: Configure Smart Software Licensing

This procedure applies for a chassis using the License Authority, Satellite server users; see the FXOS configuration guide to configure your method as a prerequisite.

Note The Strong Encryption (3DES/AES) license is not enabled by default so you cannot use ASDM to configure your ASA until you request the Strong Encryption license using the ASA CLI. Other strong encryption features are also not available until you do so.

Before you begin

For an ASA cluster, you need to access the control unit for configuration. Check the Firepower Chassis Manager to see which unit is the control unit. You can also check from the ASA CLI, as shown in this procedure.

Procedure

Step 1 Connect to the Firepower 9300 chassis CLI (console or SSH), and then session to the ASA:

connect module *slot* **console**
connect asa

Example:

```
Firepower> connect module 1 console
Firepower-module1> connect asa

asa>
```

The next time you connect to the ASA console, you go directly to the ASA; you do not need to enter **connect asa** again.

For an ASA cluster, you only need to access the control unit for license configuration and other configuration. Typically, the control unit is in slot 1, so you should connect to that module first.

Step 2 At the ASA CLI, enter global configuration mode. By default, the enable password is blank.

enable
configure terminal

Example:

```
asa> enable
Password:
asa# configure terminal
asa(config)#
```

Step 3 If required, for an ASA cluster confirm that this unit is the control unit:

show cluster info

Example:

```
asa(config)# show cluster info
Cluster stbu: On
  This is "unit-1-1" in state SLAVE
    ID : 0
    Version : 9.5(2)
    Serial No.: P3000000025
    CCL IP : 127.2.1.1
    CCL MAC : 000b.fcf8.c192
    Last join : 17:08:59 UTC Sep 26 2015
    Last leave: N/A
Other members in the cluster:
  Unit "unit-1-2" in state SLAVE
    ID : 1
    Version : 9.5(2)
    Serial No.: P3000000001
    CCL IP : 127.2.1.2
    CCL MAC : 000b.fcf8.c162
    Last join : 19:13:11 UTC Sep 23 2015
    Last leave: N/A
  Unit "unit-1-3" in state MASTER
    ID : 2
    Version : 9.5(2)
    Serial No.: JAB0815R0JY
    CCL IP : 127.2.1.3
    CCL MAC : 000f.f775.541e
    Last join : 19:13:20 UTC Sep 23 2015
    Last leave: N/A
```

If a different unit is the control unit, exit the connection and connect to the correct unit. See below for information about exiting the connection.

Step 4 Enter license smart configuration mode:

license smart

Example:

```
ciscoasa(config)# license smart
ciscoasa(config-smart-lic)#
```

Step 5 Set the feature tier:

feature tier standard

Only the standard tier is available. A tier license is a prerequisite for adding other feature licenses. You must have sufficient tier licenses in your account. Otherwise, you cannot configure any other feature licenses or any features that require licenses.

Step 6 Request one or more of the following features:

- Mobile SP (GTP/GPRS)

  feature mobile-sp

- Security Contexts

  feature context *<1-248>*

- Strong Encryption (3DES/AES)

  feature strong-encryption

Example:

```
ciscoasa(config-smart-lic)# feature strong-encryption
ciscoasa(config-smart-lic)# feature context 50
```

Step 7 To exit the ASA console, enter **~** at the prompt to exit to the Telnet application. Enter **quit** to exit back to the supervisor CLI.

Licenses Per Model

This section lists the license entitlements available for the ASAv and Firepower 9300 chassis ASA security module.

ASAv

The following table shows the licensed features for the ASAv series.

Licenses	Standard License	
Firewall Licenses		
Botnet Traffic Filter	Enabled	
Firewall Conns, Concurrent	ASAv5: 100,000 ASAv10: 100,000 ASAv30: 500,000	
GTP/GPRS	Enabled	
Total UC Proxy Sessions	ASAv5: 500 ASAv10: 500 ASAv30: 1000	
VPN Licenses		
AnyConnect peers	Unlicensed	*Optional AnyConnect Plus or Apex license, Maximums:* *ASAv5: 50* *ASAv10: 250* *ASAv30: 750*
Other VPN Peers	ASAv5: 250 ASAv10: 250 ASAv30: 1000	
Total VPN Peers, combined all types	ASAv5: 250 ASAv10: 250 ASAv30: 1000	
General Licenses		
Throughput Level	ASAv5: 1 Gbps ASAv10: 1 Gbps ASAv30: 2 Gbps	
Encryption	Strong (3DES/AES)	
Failover	Active/Standby	
Security Contexts	No support	

Licenses	Standard License
Clustering	No support
VLANs, Maximum	ASAv5: 50 ASAv10: 50 ASAv30: 200
RAM, vCPUs	ASAv5: 2 GB, 1 vCPU ASAv10: 2 GB, 1 vCPU ASAv30: 8 GB, 4 vCPUs

Firepower 9300 ASA Application

The following table shows the licensed features for the Firepower 9300 ASA application.

Licenses	Standard License	
Firewall Licenses		
Botnet Traffic Filter	No Support.	
Firewall Conns, Concurrent	Firepower 9300 SM-36: 60,000,000, up to 70,000,000 for a chassis with 3 modules Firepower 9300 SM-24: 55,000,000, up to 70,000,000 for a chassis with 3 modules	
GTP/GPRS	Disabled	*Optional License: Mobile SP*
Total UC Proxy Sessions	15,000	
VPN is not supported with Firepower Chassis Manager 1.1.2 and earlier.		
General Licenses		
Encryption	Base (DES) or Strong (3DES/AES)	
Security Contexts	10	*Optional License: Maximum of 250, in increments of 10*
Clustering	Enabled	
VLANs, Maximum	1024	

Monitoring Smart Software Licensing

You can monitor the license features, status, and certificate, as well as enable debug messages.

Viewing Your Current License

See the following commands for viewing your license:

- **show license features**

  The following example shows an ASAv with only a base license (no current license entitlement):

```
Serial Number:  9AAHGX8514R

ASAv Platform License State: Unlicensed
No active entitlement: no feature tier configured

Licensed features for this platform:
Maximum Physical Interfaces       : 10             perpetual
Maximum VLANs                     : 50             perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Standby perpetual
Encryption-DES                    : Enabled        perpetual
Encryption-3DES-AES               : Enabled        perpetual
Security Contexts                 : 0              perpetual
GTP/GPRS                          : Disabled       perpetual
AnyConnect Premium Peers          : 2              perpetual
AnyConnect Essentials             : Disabled       perpetual
Other VPN Peers                   : 250            perpetual
Total VPN Peers                   : 250            perpetual
Shared License                    : Disabled       perpetual
AnyConnect for Mobile             : Disabled       perpetual
AnyConnect for Cisco VPN Phone    : Disabled       perpetual
Advanced Endpoint Assessment      : Disabled       perpetual
UC Phone Proxy Sessions           : 2              perpetual
Total UC Proxy Sessions           : 2              perpetual
Botnet Traffic Filter             : Enabled        perpetual
Intercompany Media Engine         : Disabled       perpetual
Cluster                           : Disabled       perpetual
```

- **show license entitlement**

  Displays detailed information about each entitlement in use, its handle (i.e. integer id), its count, tag, enforcement mode (e.g. in compliance, out of compliance, etc.), version and time at which the entitlement was requested.

Viewing Smart License Status

See the following commands for viewing license status:

- **show license all**

  Displays the state of Smart Software Licensing, Smart Agent version, UDI information, Smart Agent state, global compliance status, the entitlements status, licensing certificate information, and scheduled Smart Agent tasks.

  The following example shows an ASAv license:

```
ciscoasa# show license all

Cisco Smart Licensing Agent, Version 1.1.1

Smart Licensing Enabled: Yes
```

```
UDI:
PID:ASAv,SN:9AC5KH5H9FW

Compliance Status: In Compliance

Assigned License Pool: ASAv Internal Users

Grace period: Not in use

Entitlement:
    Tag: regid.2014-08.com.cisco.ASAv-STD-1G,1.0_4fd3bdbd-29ae-4cce-ad82-45ad3db1070c,
 Version: 1.0, Enforce Mode: Authorized
    Requested Time: Sep 15 13:08:23 2015 UTC,  Requested Count: 1
    Vendor String: (null)

Smart Licensing State: authorized (4)

Licensing Certificates:
    ID Cert Info:
        Start Date: Sep 15 12:59:29 2015 UTC. Expiry Date: Sep 14 12:59:29 2016 UTC
        Serial Number: 214929
        Version: 3
        Subject/SN: 16cab27f-a239-4d2d-a8db-d81dc48ec6bb
        Common Name: 55d246e3160a5dab39ec218f0b6b00f03422ef0d::1,2
    Signing Cert Info:
        Start Date: Jun 14 20:18:52 2013 UTC. Expiry Date: Apr 24 21:55:42 2033 UTC
        Serial Number: 3
        Version: 3

Upcoming Scheduled Jobs:
    Certificate Renewal: Mar 13 13:03:06 2016 UTC (172 days, 11 hours, 24 minutes, 8
seconds remaining)
    Certificate Expiration: Sep 14 13:00:03 2016 UTC (357 days, 11 hours, 21 minutes,
5 seconds remaining)
    Authorization Renewal: Oct 22 18:58:18 2015 UTC (29 days, 17 hours, 19 minutes, 20
 seconds remaining)
    Authorization Expiration: Dec 21 18:55:28 2015 UTC (89 days, 17 hours, 16 minutes,
 30 seconds remaining)
    Daily Job: Sep 23 13:09:27 2015 UTC (11 hours, 30 minutes, 29 seconds remaining)

HA Info:    HA not available
    HA Sudi: Not Available
```

- **show license registration**

  Displays the current Smart License registration status.

- **show license pool**

  Displays the entitlement pool to which this device is assigned.

Displaying ID Certificate Information

See the following command to view the license ID certificate:

- **show license cert**

  Displays the ID certificate content, date issued, and the date it expires.

Debugging Smart Software Licensing

See the following commands for debugging clustering:

- **debug license agent** {**error** | **trace** | **debug** | **all**}

  Turns on debugging from the Smart Agent.

- **debug license** *level*

  Turns on various levels of Smart Software Licensing Manager debugs.

Smart Software Manager Communication

This section describes how your device communicates with the Smart Software Manager.

Device Registration and Tokens

For each virtual account, you can create a registration token. This token is valid for 30 days by default. Enter this token ID plus entitlement levels when you deploy each device, or when you register an existing device. You can create a new token if an existing token is expired.

Note Firepower 9300 chassis—Device registration is configured in the chassis, not on the ASA logical device.

At startup after deployment, or after you manually configure these parameters on an existing device, the device registers with the Cisco License Authority. When the device registers with the token, the License Authority issues an ID certificate for communication between the device and the License Authority. This certificate is valid for 1 year, although it will be renewed every 6 months.

Periodic Communication with the License Authority

The device communicates with the License Authority every 30 days. If you make changes in the Smart Software Manager, you can refresh the authorization on the device so the change takes place immediately. Or you can wait for the device to communicate as scheduled.

You can optionally configure an HTTP proxy.

ASAv

The ASAv must have internet access either directly or through an HTTP proxy at least every 30 days. The ASAv does not have a grace period. You must contact the Licensing Authority, or the ASAv will be severely rate-limited until you are able to successfully reauthorize.

Firepower 9300

The Firepower 9300 must have internet access either directly or through an HTTP proxy at least every 90 days. Normal license communication occurs every 30 days, but with the grace period, your device will operate for up to 90 days without calling home. After the grace period, you must contact the Licensing Authority, or you will not be able to make configuration changes to features requiring special licenses; operation is otherwise unaffected.

Out-of-Compliance State

The device can become out of compliance in the following situations:

- Over-utilization—When the device uses unavailable licenses.
- License expiration—When a time-based license expires.
- Lack of communication—When the device cannot reach the Licensing Authority for re-authorization.

To verify whether your account is in, or approaching, an Out-of-Compliance state, you must compare the entitlements currently in use by your device against those in your Smart Account.

In an out-of-compliance state, the device might be limited, depending on the model:

- ASAv—The ASAv will be severely rate-limited until you are able to successfully reauthorize.
- Firepower 9300—You will not be able to make configuration changes to features requiring special licenses, but operation is otherwise unaffected. For example, existing contexts over the Standard license limit can continue to run, and you can modify their configuration, but you will not be able to add a *new* context. If you do not have sufficient Standard licenses when you first register, you cannot configure any licensed features, including strong encryption features.

Smart Call Home Infrastructure

By default, a Smart Call Home profile exists in the configuration that specifies the URL for the Licensing Authority. You cannot remove this profile. Note that the only configurable option for the License profile is the destination address URL for the License Authority. Unless directed by Cisco TAC, you should not change the License Authority URL.

Note For the Firepower 9300 chassis, Smart Call Home for licensing is configured in the Firepower 9300 chassis supervisor, not on the ASA.

You cannot disable Smart Call Home for Smart Software Licensing. For example, even if you disable Smart Call Home using the **no service call-home** command, Smart Software Licensing is not disabled.

Other Smart Call Home functions are not turned on unless you specifically configure them.

History for Smart Software Licensing

Feature Name	Platform Releases	Description
Cisco Smart Software Licensing for the ASA on the Firepower 9300	9.4(1.150)	We introduced Smart Software Licensing for the ASA on the Firepower 9300. We introduced the following commands: **feature strong-encryption, feature mobile-sp, feature context**
Cisco Smart Software Licensing for the ASAv	9.3(2)	Smart Software Licensing lets you purchase and manage a pool of licenses. Unlike PAK licenses, smart licenses are not tied to a specific serial number. You can easily deploy or retire ASAvs without having to manage each unit's license key. Smart Software Licensing also lets you see your license usage and needs at a glance. We introduced the following commands: **clear configure license, debug license agent, feature tier, http-proxy, license smart, license smart deregister, license smart register, license smart renew, show license, show running-config license, throughput level**

CHAPTER 5

Logical Devices for the Firepower 9300

The Firepower 9300 is a flexible security platform on which you can install one or more *logical devices*. This chapter describes basic interface configuration and how to add a standalone or High Availability logical device using the Firepower Chassis Manager. To add a clustered logical device, see ASA Cluster for the Firepower 9300 Chassis, on page 357. To use the FXOS CLI, see the FXOS CLI configuration guide. For more advanced FXOS procedures and troubleshooting, see the FXOS configuration guide.

About Firepower Interfaces

The Firepower 9300 chassis supports physical interfaces and EtherChannel (port-channel) interfaces. EtherChannel interfaces can include up to 16 member interfaces of the same type.

Chassis Management Interface

The chassis management interface is used for management of the FXOS Chassis by SSH or Firepower Chassis Manager. This interface is separate from the mgmt-type interface that you assign to the logical devices for application management.

To configure parameters for this interface, you must configure them from the CLI. To view information about this interface in the FXOS CLI, connect to local management and show the management port:

Firepower # **connect local-mgmt**

Firepower(local-mgmt) # **show mgmt-port**

Note that the chassis management interface remains up even if the physical cable or SFP module are unplugged, or if the **mgmt-port shut** command is performed.

Note The chassis management interface does not support jumbo frames.

Interface Types

Each interface can be one of the following types:

- Data—Use for regular data. Data interfaces cannot be shared between logical devices, and logical devices cannot communicate over the backplane to other logical devices. For traffic on Data interfaces, all traffic must exit the chassis on one interface and return on another interface to reach another logical device.
- Mgmt—Use to manage application instances. These interfaces can be shared by one or more logical devices to access external hosts; logical devices cannot communicate over this interface with other logical devices that share the interface. You can only assign one management interface per logical device. For ASA: You can later enable management from a data interface; but you must assign a Management interface to the logical device even if you don't intend to use it after you enable data management. For information about the separate chassis management interface, see Chassis Management Interface, on page 131.
- Firepower-eventing—Use as a secondary management interface for FTD devices.
- Cluster—Use as the cluster control link for a clustered logical device. By default, the cluster control link is automatically created on Port-channel 48. The Cluster type is only supported on EtherChannel interfaces.

FXOS Interfaces vs. Application Interfaces

The Firepower 9300 manages the basic Ethernet settings of physical interfaces and EtherChannel (port-channel) interfaces. Within the application, you configure higher level settings. For example, you can only create EtherChannels in FXOS; but you can assign an IP address to the EtherChannel within the application.

The following sections describe the interaction between FXOS and the application for interfaces.

VLAN Subinterfaces

For all logical devices, you can create VLAN subinterfaces within the application.

Independent Interface States in the Chassis and in the Application

You can administratively enable and disable interfaces in both the chassis and in the application. For an interface to be operational, the interface must be enabled in both operating systems. Because the interface state is controlled independently, you may have a mismatch between the chassis and application.

About Logical Devices

A logical device lets you run one application instance.

When you add a logical device, you also define the application instance type and version, assign interfaces, and configure bootstrap settings that are pushed to the application configuration.

Standalone and Clustered Logical Devices

You can add the following logical device types:

- Standalone—A standalone logical device operates as a standalone unit or as a unit in a High Availability pair.
- Cluster—A clustered logical device lets you group multiple units together, providing all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices. Multiple module devices, like the Firepower 9300, support intra-chassis clustering. For the Firepower 9300, all three modules must participate in the cluster.

Requirements and Prerequisites for Hardware and Software Combinations

The Firepower 9300 supports multiple models, security modules, application types, and high availability and scalability features. See the following requirements for allowed combinations.

Firepower 9300 Requirements

The Firepower 9300 includes 3 security module slots and multiple types of security modules. See the following requirements:

- Security Module Types—All modules in the Firepower 9300 must be the same type.
- Clustering—All security modules in the cluster, whether it is intra-chassis or inter-chassis, must be the same type. You can have different quantities of installed security modules in each chassis, although all modules present in the chassis must belong to the cluster including any empty slots. For example, you can install 2 SM-36s in chassis 1, and 3 SM-36s in chassis 2.
- High Availability—High Availability is only supported between same-type modules on the Firepower 9300.
- ASA and FTD application types—You can only install one application type on the chassis, ASA or FTD.
- ASA or FTD versions—You can run different versions of an application instance type on separate modules. For example, you can install FTD 6.3 on module 1, FTD 6.4 on module 2, and FTD 6.5 on module 3.

Guidelines and Limitations for Logical Devices

See the following sections for guidelines and limitations.

Guidelines and Limitations for Firepower Interfaces

Default MAC Addresses

Default MAC address assignments depend on the type of interface.

- Physical interfaces—The physical interface uses the burned-in MAC address.
- EtherChannels—For an EtherChannel, all interfaces that are part of the channel group share the same MAC address. This feature makes the EtherChannel transparent to network applications and users, because they only see the one logical connection; they have no knowledge of the individual links. The port-channel interface uses a unique MAC address from a pool; interface membership does not affect the MAC address.

General Guidelines and Limitations

Firewall Mode

You can set the firewall mode to routed or transparent in the bootstrap configuration for the FTD. For the ASA, you can change the firewall mode to transparent after you deploy. See Change the ASA to Transparent Firewall Mode, on page 144.

High Availability

- Configure high availability within the application configuration.
- You can use any data interfaces as the failover and state links.

Context Mode

- Enable multiple context mode in the ASA after you deploy.

Requirements and Prerequisites for High Availability

- The two units in a High Availability Failover configuration must:
  - Be on a separate chassis; intra-chassis High Availability for the Firepower 9300 is not supported.
  - Be the same model.
  - Have the same interfaces assigned to the High Availability logical devices.
  - Have the same number and types of interfaces. All interfaces must be preconfigured in FXOS identically before you enable High Availability.
- For High Availability system requirements, see Failover System Requirements, on page 216.

Configure Interfaces

By default, physical interfaces are disabled. You can enable interfaces, add EtherChannels, and edit interface properties.

Note If you remove an interface in FXOS (for example, if you remove a network module, remove an EtherChannel, or reassign an interface to an EtherChannel), then the ASA configuration retains the original commands so that you can make any necessary adjustments; removing an interface from the configuration can have wide effects. You can manually remove the old interface configuration in the ASA OS.

Configure a Physical Interface

You can physically enable and disable interfaces, as well as set the interface speed and duplex. To use an interface, it must be physically enabled in FXOS and logically enabled in the application.

Before you begin

- Interfaces that are already a member of an EtherChannel cannot be modified individually. Be sure to configure settings before you add it to the EtherChannel.

Procedure

Step 1 Enter interface mode.

scope eth-uplink

scope fabric a

Step 2 Enable the interface.

enter interface *interface_id*

enable

Example:

```
Firepower /eth-uplink/fabric # enter interface Ethernet1/8
Firepower /eth-uplink/fabric/interface # enable
```

Note Interfaces that are already a member of a port-channel cannot be modified individually. If you use the **enter interface** or **scope interface** command on an interface that is a member of a port channel, you will receive an error stating that the object does not exist. You should edit interfaces using the **enter interface** command before you add them to a port-channel.

Step 3 (Optional) Set the interface type.

set port-type {**data** | **mgmt** | **cluster**}

Example:

```
Firepower /eth-uplink/fabric/interface # set port-type mgmt
```

The **data** keyword is the default type. Do not choose the **cluster** keyword; by default, the cluster control link is automatically created on Port-channel 48.

Step 4 Enable or disable autonegotiation, if supported for your interface.

set auto-negotiation {**on** | **off**}

Example:

```
Firepower /eth-uplink/fabric/interface* # set auto-negotiation off
```

Step 5 Set the interface speed.

set admin-speed {**10mbps** | **100mbps** | **1gbps** | **10gbps** | **40gbps** | **100gbps**}

Example:

```
Firepower /eth-uplink/fabric/interface* # set admin-speed 1gbps
```

Step 6 Set the interface duplex mode.

set admin-duplex {**fullduplex** | **halfduplex**}

Example:

```
Firepower /eth-uplink/fabric/interface* # set admin-duplex halfduplex
```

Step 7 If you edited the default flow control policy, it is already applied to interfaces. If you created a new policy, apply it to the interface.

set flow-control-policy *name*

Example:

```
Firepower /eth-uplink/fabric/interface* # set flow-control-policy flow1
```

Step 8 Save the configuration.

commit-buffer

Example:

```
Firepower /eth-uplink/fabric/interface* # commit-buffer
Firepower /eth-uplink/fabric/interface #
```

Add an EtherChannel (Port Channel)

An EtherChannel (also known as a port channel) can include up to 16 member interfaces of the same media type and capacity, and must be set to the same speed and duplex. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface. The Link Aggregation Control Protocol (LACP) aggregates interfaces by exchanging the Link Aggregation Control Protocol Data Units (LACPDUs) between two network devices.

The Firepower 9300 chassis only supports EtherChannels in Active LACP mode so that each member interface sends and receives LACP updates. An active EtherChannel can establish connectivity with either an active or a passive EtherChannel. You should use the active mode unless you need to minimize the amount of LACP traffic.

LACP coordinates the automatic addition and deletion of links to the EtherChannel without user intervention. It also handles misconfigurations and checks that both ends of member interfaces are connected to the correct channel group.

Procedure

Step 1 Enter interface mode:

scope eth-uplink

scope fabric a

Step 2 Create the port-channel:

create port-channel *id*

enable

Step 3 Assign member interfaces:

create member-port *interface_id*

You can add up to 16 member interfaces of the same media type and capacity. The member interfaces must be set to the same speed and duplex, and must match the speed and duplex that you configured for this port channel. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface.

Example:

```
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
```

Step 4 (Optional) Set the interface type.

set port-type {**data** | **mgmt** | **cluster**}

Example:

```
Firepower /eth-uplink/fabric/port-channel # set port-type data
```

The **data** keyword is the default type. Do not choose the **cluster** keyword unless you want to use this port-channel as the cluster control link instead of the default.

Step 5 Set the required interface speed for members of the port-channel.

set speed {**10mbps** | **100mbps** | **1gbps** | **10gbps** | **40gbps** | **100gbps**}

If you add a member interface that is not at the specified speed, it will not successfully join the port channel. The default is **10gbps**.

Example:

```
Firepower /eth-uplink/fabric/port-channel* # set speed 1gbps
```

Step 6 (Optional) Set the required duplex for members of the port-channel.

set duplex {**fullduplex** | **halfduplex**}

If you add a member interface that is configured with the specified duplex, it will not successfully join the port channel. The default is **fullduplex**.

Example:

```
Firepower /eth-uplink/fabric/port-channel* # set duplex fullduplex
```

Step 7 Enable or disable autonegotiation, if supported for your interface.

set auto-negotiation {**on** | **off**}

Example:

```
Firepower /eth-uplink/fabric/interface* # set auto-negotiation off
```

Step 8 If you edited the default flow control policy, it is already applied to interfaces. If you created a new policy, apply it to the interface.

set flow-control-policy *name*

Example:

```
Firepower /eth-uplink/fabric/interface* # set flow-control-policy flow1
```

Step 9 Commit the configuration:

commit-buffer

Configure Logical Devices

Add a standalone logical device or a High Availability pair on the Firepower 9300 chassis.

For clustering, see #unique_211.

Add a Standalone ASA

Standalone logical devices work either alone or in a High Availability pair. On the Firepower 9300 with multiple security modules, you can deploy either a cluster or standalone devices. The cluster must use all modules, so you cannot mix and match a 2-module cluster plus a single standalone device, for example.

You can deploy a routed firewall mode ASA from the Firepower 9300 chassis. To change the ASA to transparent firewall mode, complete this procedure, and then see Change the ASA to Transparent Firewall Mode, on page 144.

For multiple context mode, you must first deploy the logical device, and then enable multiple context mode in the ASA application.

Before you begin

- Download the application image you want to use for the logical device from Cisco.com, and then download that image to the Firepower 9300 chassis.
- Configure a management interface to use with the logical device. The management interface is required. Note that this management interface is not the same as the chassis management port that is used only for chassis management (in FXOS, you might see it displayed as MGMT, management0, or other similar names).
- Gather the following information:
  - Interface IDs for this device
  - Management interface IP address and network mask
  - Gateway IP address

Procedure

Step 1 Enter security services mode.

scope ssa

Example:

```
Firepower# scope ssa
Firepower /ssa #
```

Step 2 Set the application instance image version.

a) View available images. Note the Version number that you want to use.

show app

Example:

```
Firepower /ssa # show app
    Name       Version         Author     Supported Deploy Types CSP Type    Is Default
 App
    ---------- --------------- ---------- ---------------------- -----------
--------------
    asa        9.9.1           cisco      Native                 Application No
    asa        9.10.1          cisco      Native                 Application Yes
    ftd        6.2.3           cisco      Native                 Application Yes
```

b) Set the scope to the security module/engine slot.

scope slot *slot_id*

The *slot_id* is 1, 2, or 3 for the Firepower 9300.

Example:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot #
```

c) Create the application instance.

enter app-instance asa

Example:

```
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* #
```

d) Set the ASA image version.

set startup-version *version*

Example:

```
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
```

e) Exit to slot mode.

exit

Example:

```
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* #
```

f) Exit to ssa mode.

exit

Example:

```
Firepower /ssa/slot* # exit
Firepower /ssa* #
```

Example:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* # exit
Firepower /ssa* #
```

Step 3 Create the logical device.

enter logical-device *device_name* **asa** *slot_id* **standalone**

Example:

```
Firepower /ssa # enter logical-device ASA1 asa 1 standalone
Firepower /ssa/logical-device* #
```

Step 4 Assign the management and data interfaces to the logical device. Repeat for each interface.

create external-port-link *name interface_id* **asa**

set description *description*

exit

- *name*—The name is used by the Firepower 9300 chassis supervisor; it is not the interface name used in the ASA configuration.
- *description*—Use quotes (") around phrases with spaces.

The management interface is not the same as the chassis management port. You will later enable and configure the data interfaces on the ASA, including setting the IP addresses.

Example:

```
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 asa
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 asa
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 asa
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
```

Step 5 Configure the management bootstrap information.

a) Create the bootstrap object.

   create mgmt-bootstrap asa

   Example:

```
Firepower /ssa/logical-device* # create mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

b) Specify the admin password.

   create bootstrap-key-secret PASSWORD

   set value

   Enter a value: *password*

Confirm the value: *password*

exit

Example:

The pre-configured ASA admin user is useful for password recovery; if you have FXOS access, you can reset the admin user password if you forget it.

Example:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Enter a value: floppylampshade
Confirm the value: floppylampshade
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

c) Configure the IPv4 management interface settings.

create ipv4 *slot_id* **default**

set ip *ip_address* **mask** *network_mask*

set gateway *gateway_address*

exit

Example:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 default
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.10.10.34 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.10.10.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

d) Configure the IPv6 management interface settings.

create ipv6 *slot_id* **default**

set ip *ip_address* **prefix-length** *prefix*

set gateway *gateway_address*

exit

Example:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv6 1 default
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # set ip 2001:0DB8:BA98::3210
prefix-length 64
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # set gateway 2001:0DB8:BA98::3211
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

e) Exit the management bootstrap mode.

exit

Example:

```
   Firepower /ssa/logical-device/mgmt-bootstrap* # exit
   Firepower /ssa/logical-device* #
```

Step 6 Save the configuration.

commit-buffer

The chassis deploys the logical device by downloading the specified software version and pushing the bootstrap configuration and management interface settings to the application instance. Check the status of the deployment using the **show app-instance** command. The application instance is running and ready to use when the **Admin State** is **Enabled** and the **Oper State** is **Online**.

Example:

```
Firepower /ssa/logical-device* # commit-buffer
Firepower /ssa/logical-device # exit
Firepower /ssa # show app-instance
App Name   Identifier Slot ID    Admin State Oper State       Running Version Startup Version
 Deploy Type Profile Name Cluster State   Cluster Role
---------- ---------- ---------- ----------- ---------------- --------------- ---------------
 ----------- ------------ --------------- ------------
asa        asa1       2          Disabled    Not Installed                    9.12.1
    Native                   Not Applicable  None
ftd        ftd1       1          Enabled     Online           6.4.0.49        6.4.0.49
    Container   Default-Small Not Applicable  None
```

Step 7 See the ASA configuration guide to start configuring your security policy.

Example

```
Firepower# scope ssa
Firepower /ssa # scope slot 1
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* # exit
Firepower /ssa* # create logical-device MyDevice1 asa 1 standalone
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 asa
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 asa
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 asa
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Enter a value: secretglassine
Confirm the value: secretglassine
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 default
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.31 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # commit-buffer
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key #
```

Add a High Availability Pair

ASA High Availability (also known as failover) is configured within the application, not in FXOS. However, to prepare your chassis for high availability, see the following steps.

Before you begin

See Failover System Requirements, on page 216.

Procedure

Step 1 Allocate the same interfaces to each logical device.

Step 2 Allocate 1 or 2 data interfaces for the failover and state link(s).

These interfaces exchange high availability traffic between the 2 chassis. We recommend that you use a 10 GB data interface for a combined failover and state link. If you have available interfaces, you can use separate failover and state links; the state link requires the most bandwidth. You cannot use the management-type interface for the failover or state link. We recommend that you use a switch between the chassis, with no other device on the same network segment as the failover interfaces.

Step 3 Enable High Availability on the logical devices. See Failover for High Availability, on page 215.

Step 4 If you need to make interface changes after you enable High Availability, perform the changes on the standby unit first, and then perform the changes on the active unit.

Note For the ASA, if you remove an interface in FXOS (for example, if you remove a network module, remove an EtherChannel, or reassign an interface to an EtherChannel), then the ASA configuration retains the original commands so that you can make any necessary adjustments; removing an interface from the configuration can have wide effects. You can manually remove the old interface configuration in the ASA OS.

Change the ASA to Transparent Firewall Mode

You can only deploy a routed firewall mode ASA from the Firepower 9300 chassis. To change the ASA to transparent firewall mode, complete the initial deployment, and then change the firewall mode within the ASA CLI. For standalone ASAs, because changing the firewall mode erases the configuration, you must then redeploy the configuration from the Firepower 9300 chassis to regain the bootstrap configuration. The ASA then remains in transparent mode with a working bootstrap configuration. For clustered ASAs, the configuration is not erased, so you do not need to redeploy the bootstrap configuration from FXOS.

Procedure

Step 1 Connect to the ASA console according to Connect to the Console of the Application, on page 147. For a cluster, connect to the primary unit. For a failover pair, connect to the active unit.

Step 2 Enter configuration mode:

enable

configure terminal

By default, the enable password is blank.

Step 3 Set the firewall mode to transparent:

firewall transparent

Step 4 Save the configuration:

write memory

For a cluster or failover pair, this configuration is replicated to secondary units:

```
asa(config)# firewall transparent
asa(config)# write memory
Building configuration...
Cryptochecksum: 9f831dfb 60dffa8c 1d939884 74735b69

3791 bytes copied in 0.160 secs
[OK]
asa(config)#
Beginning configuration replication to unit-1-2
 End Configuration Replication to data unit.

asa(config)#
```

Step 5 On the Firepower Chassis Manager **Logical Devices** page, click the **Edit** icon to edit the ASA.

The **Provisioning** page appears.

Step 6 Click the device icon to edit the bootstrap configuration. Change any value in your configuration, and click **OK**.

You must change the value of at least one field, for example, the **Password** field.

You see a warning about changing the bootstrap configuration; click **Yes**.

Step 7 Click **Save** to redeploy the configuration to the ASA.

Wait several minutes for the chassis/security modules to reload, and for the ASA to become operational again. The ASA now has an operational bootstrap configuration, but remains in transparent mode.

Change an Interface on an ASA Logical Device

You can allocate, unallocate, or replace a management interface on an ASA logical device. ASDM discovers the new interfaces automatically.

Adding a new interface, or deleting an unused interface has minimal impact on the ASA configuration. However, if you remove an allocated interface in FXOS (for example, if you remove a network module, remove an EtherChannel, or reassign an allocated interface to an EtherChannel), and the interface is used in your security policy, removal will impact the ASA configuration. In this case, the ASA configuration retains the original commands so that you can make any necessary adjustments. You can manually remove the old interface configuration in the ASA OS.

Note You can edit the membership of an allocated EtherChannel without impacting the logical device.

Before you begin

- Configure your interfaces and add any EtherChannels according to Configure a Physical Interface, on page 135 and Add an EtherChannel (Port Channel), on page 136.
- If you want to add an already-allocated interface to an EtherChannel (for example, all interfaces are allocated by default to a cluster), you need to unallocate the interface from the logical device first, then add the interface to the EtherChannel. For a new EtherChannel, you can then allocate the EtherChannel to the device.
- For clustering or failover, make sure you add or remove the interface on all units. We recommend that you make the interface changes on the data/standby unit(s) first, and then on the control/active unit. New interfaces are added in an administratively down state, so they do not affect interface monitoring.

Procedure

Step 1 Enter security services mode:

Firepower# **scope ssa**

Step 2 Edit the logical device:

Firepower /ssa # **scope logical-device** *device_name*

Step 3 Unallocate an interface from the logical device:

Firepower /ssa/logical-device # **delete external-port-link** *name*

Enter the **show external-port-link** command to view interface names.

For a management interface, delete the current interface then commit your change using the **commit-buffer** command before you add the new management interface.

Step 4 Allocate a new interface to the logical device:

Firepower /ssa/logical-device* # **create external-port-link** *name interface_id* **asa**

Step 5 Commit the configuration:

commit-buffer

Commits the transaction to the system configuration.

Connect to the Console of the Application

Use the following procedure to connect to the console of the application.

Procedure

Step 1 Connect to the module CLI.

connect module *slot_number* **console**

To connect to the security engine of a device that does not support multiple security modules, always use **1** as the *slot_number*.

Example:

```
Firepower# connect module 1 console
Telnet escape character is '~'.
Trying 127.5.1.1...
Connected to 127.5.1.1.
Escape character is '~'.

CISCO Serial Over LAN:
Close Network Connection to Exit

Firepower-module1>
```

Step 2 Connect to the application console.

connect asa

Example:

```
Firepower-module1> connect asa
Connecting to asa(asa1) console... hit Ctrl + A + D  to return to bootCLI
[...]
asa>
```

Step 3 Exit the application console to the FXOS module CLI.

- ASA—Enter **Ctrl-a, d**

Step 4 Return to the supervisor level of the FXOS CLI.

a) Enter ~

You exit to the Telnet application.

b) To exit the Telnet application, enter:

telnet>**quit**

History for Logical Devices

Feature	Version	Details
Intra-chassis ASA Clustering for the Firepower 9300	9.4(1.150)	You can cluster up to 3 security modules within the Firepower 9300 chassis. All modules in the chassis must belong to the cluster. We introduced the following commands: **cluster replication delay, debug service-module, management-only individual, show cluster chassis**

CHAPTER 6

Transparent or Routed Firewall Mode

This chapter describes how to set the firewall mode to routed or transparent, as well as how the firewall works in each firewall mode.

You can set the firewall mode independently for each context in multiple context mode.

About the Firewall Mode

The ASA supports two firewall modes: Routed Firewall mode and Transparent Firewall mode.

About Routed Firewall Mode

In routed mode, the ASA is considered to be a router hop in the network. Each interface that you want to route between is on a different subnet. You can share Layer 3 interfaces between contexts.

About Transparent Firewall Mode

Traditionally, a firewall is a routed hop and acts as a default gateway for hosts that connect to one of its screened subnets. A transparent firewall, on the other hand, is a Layer 2 firewall that acts like a “bump in the wire,” or a “stealth firewall,” and is not seen as a router hop to connected devices. However, like any other firewall, access control between interfaces is controlled, and all of the usual firewall checks are in place.

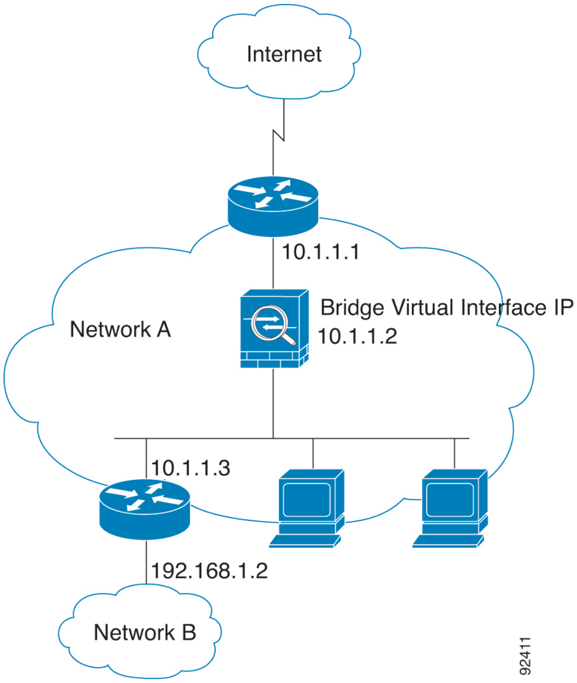
Layer 2 connectivity is achieved by using a "bridge group" where you group together the inside and outside interfaces for a network, and the ASA uses bridging techniques to pass traffic between the interfaces. Each bridge group includes a Bridge Virtual Interface (BVI) to which you assign an IP address on the network. You can have multiple bridge groups for multiple networks. In transparent mode, these bridge groups cannot communicate with each other.

Using the Transparent Firewall in Your Network

The ASA connects the same network between its interfaces. Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network.

The following figure shows a typical transparent firewall network where the outside devices are on the same subnet as the inside devices. The inside router and hosts appear to be directly connected to the outside router.

Figure 13: Transparent Firewall Network



About Bridge Groups

A bridge group is a group of interfaces that the ASA bridges instead of routes. Bridge groups are only supported in Transparent Firewall Mode. Like any other firewall interfaces, access control between interfaces is controlled, and all of the usual firewall checks are in place.

Bridge Virtual Interface (BVI)

Each bridge group includes a Bridge Virtual Interface (BVI). The ASA uses the BVI IP address as the source address for packets originating from the bridge group. The BVI IP address must be on the same subnet as the bridge group member interfaces. The BVI does not support traffic on secondary networks; only traffic on the same network as the BVI IP address is supported.

Only bridge group member interfaces are named and can be used with interface-based features.

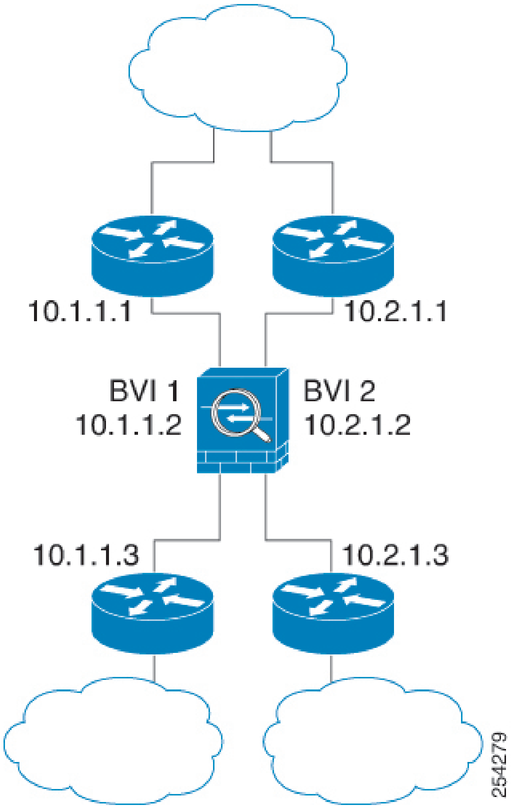
Bridge Groups in Transparent Firewall Mode

Bridge group traffic is isolated from other bridge groups; traffic is not routed to another bridge group within the ASA, and traffic must exit the ASA before it is routed by an external router back to another bridge group in the ASA. Although the bridging functions are separate for each bridge group, many other functions are shared between all bridge groups. For example, all bridge groups share a syslog server or AAA server configuration. For complete security policy separation, use security contexts with one bridge group in each context.

You can include multiple interfaces per bridge group. See Guidelines for Firewall Mode, on page 155 for the exact number of bridge groups and interfaces supported. If you use more than 2 interfaces per bridge group, you can control communication between multiple segments on the same network, and not just between inside and outside. For example, if you have three inside segments that you do not want to communicate with each other, you can put each segment on a separate interface, and only allow them to communicate with the outside interface. Or you can customize the access rules between interfaces to allow only as much access as desired.

The following figure shows two networks connected to the ASA, which has two bridge groups.

Figure 14: Transparent Firewall Network with Two Bridge Groups



Management Interface

In addition to each Bridge Virtual Interface (BVI) IP address, you can add a separate Management *slot*/*port* interface that is not part of any bridge group, and that allows only management traffic to the ASA. For more information, see Management Interface, on page 402.

Allowing Layer 3 Traffic

- Unicast IPv4 and IPv6 traffic is allowed through the bridge group automatically from a higher security interface to a lower security interface, without an access rule.
- For Layer 3 traffic traveling from a low to a high security interface, an access rule is required on the low security interface.

- ARPs are allowed through the bridge group in both directions without an access rule. ARP traffic can be controlled by ARP inspection.
- IPv6 neighbor discovery and router solicitation packets can be passed using access rules.
- Broadcast and multicast traffic can be passed using access rules.

Allowed MAC Addresses

The following destination MAC addresses are allowed through the bridge group if allowed by your access policy (see Allowing Layer 3 Traffic, on page 151). Any MAC address not on this list is dropped.

- TRUE broadcast destination MAC address equal to FFFF.FFFF.FFFF
- IPv4 multicast MAC addresses from 0100.5E00.0000 to 0100.5EFE.FFFF
- IPv6 multicast MAC addresses from 3333.0000.0000 to 3333.FFFF.FFFF
- BPDU multicast address equal to 0100.0CCC.CCCD
- AppleTalk multicast MAC addresses from 0900.0700.0000 to 0900.07FF.FFFF

Passing Traffic Not Allowed in Routed Mode

In routed mode, some types of traffic cannot pass through the ASA even if you allow it in an access rule. The bridge group, however, can allow almost any traffic through using either an access rule (for IP traffic) or an EtherType rule (for non-IP traffic):

- IP traffic—In routed firewall mode, broadcast and multicast traffic is blocked even if you allow it in an access rule, including unsupported dynamic routing protocols and DHCP (unless you configure DHCP relay). Within a bridge group, you can allow this traffic with an access rule (using an extended ACL).
- Non-IP traffic—AppleTalk, IPX, BPDUs, and MPLS, for example, can be configured to go through using an EtherType rule.

Note The bridge group does not pass CDP packets packets, or any packets that do not have a valid EtherType greater than or equal to 0x600. An exception is made for BPDUs and IS-IS, which are supported.

BPDU Handling

To prevent loops using the Spanning Tree Protocol, BPDUs are passed by default. To block BPDUs, you need to configure an EtherType rule to deny them. You can also block BPDUs on the external switches. For example, you can block BPDUs on the switch if members of the same bridge group are connected to switch ports in different VLANs. In this case, BPDUs from one VLAN will be visible in the other VLAN, which can cause Spanning Tree Root Bridge election process problems.

If you are using failover, you might want to block BPDUs to prevent the switch port from going into a blocking state when the topology changes. See Transparent Firewall Mode Bridge Group Requirements for Failover, on page 228 for more information.

MAC Address vs. Route Lookups

For traffic within a bridge group, the outgoing interface of a packet is determined by performing a destination MAC address lookup instead of a route lookup.

Route lookups, however, are necessary for the following situations:

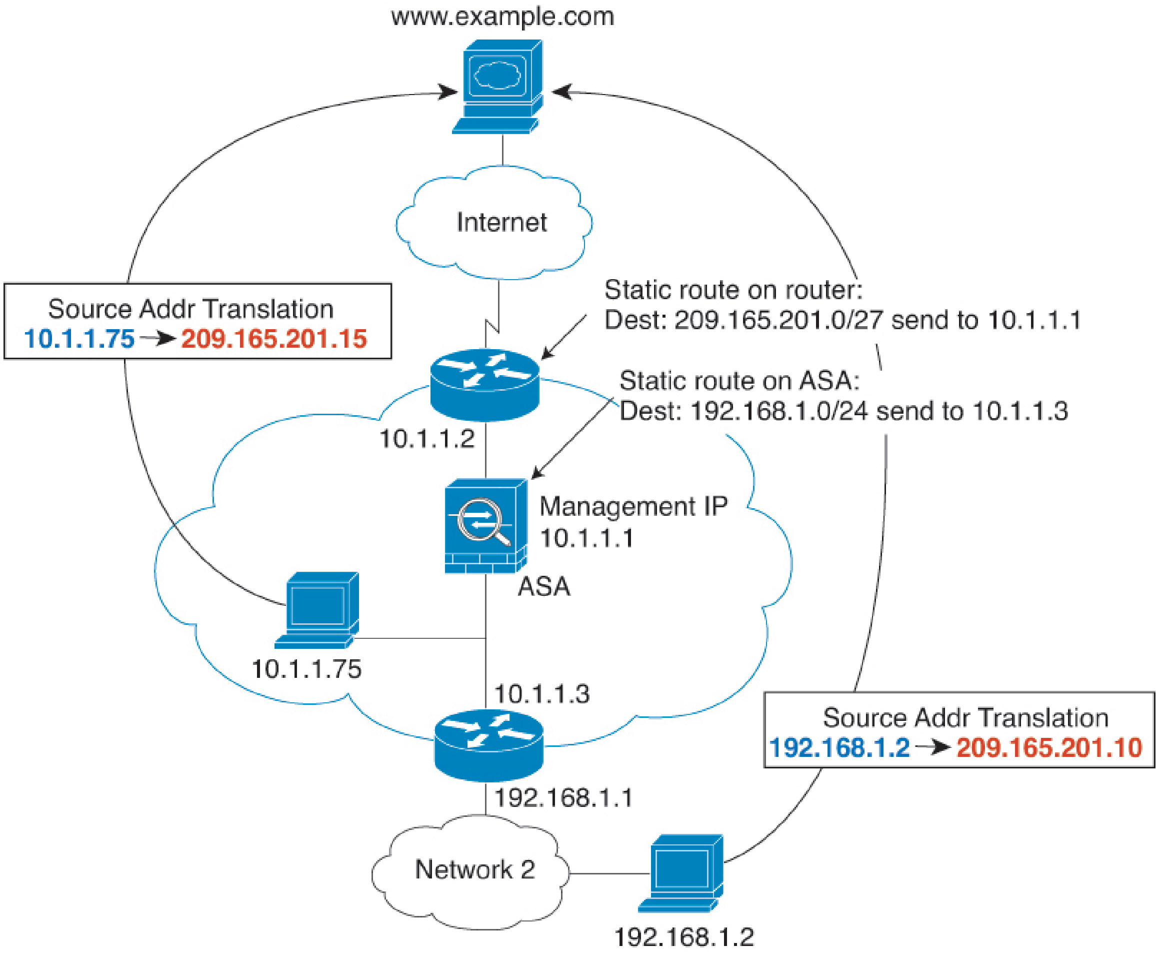
- Traffic originating on the ASA—Add a default/static route on the ASA for traffic destined for a remote network where a syslog server, for example, is located.
- Voice over IP (VoIP) and TFTP traffic with inspection enabled, and the endpoint is at least one hop away—Add a static route on the ASA for traffic destined for the remote endpoint so that secondary connections are successful. The ASA creates a temporary "pinhole" in the access control policy to allow the secondary connection; and because the connection might use a different set of IP addresses than the primary connection, the ASA needs to perform a route lookup to install the pinhole on the correct interface.

  Affected applications include:

  - CTIQBE
  - GTP
  - H.323
  - MGCP
  - RTSP
  - SIP
  - Skinny (SCCP)
  - SQL*Net
  - SunRPC
  - TFTP
- Traffic at least one hop away for which the ASA performs NAT—Configure a static route on the ASA for traffic destined for the remote network. You also need a static route on the upstream router for traffic destined for the mapped addresses to be sent to the ASA.

  This routing requirement is also true for embedded IP addresses for VoIP and DNS with inspection and NAT enabled, and the embedded IP addresses are at least one hop away. The ASA needs to identify the correct egress interface so it can perform the translation.

Figure 15: NAT Example: NAT within a Bridge Group



Unsupported Features for Bridge Groups in Transparent Mode

The following table lists the features are not supported in bridge groups in transparent mode.

Table 3: Unsupported Features in Transparent Mode

Feature	Description
Dynamic DNS	—
DHCP relay	The transparent firewall can act as a DHCPv4 server, but it does not support DHCP relay. DHCP relay is not required because you can allow DHCP traffic to pass through using two access rules: one that allows DCHP requests from the inside interface to the outside, and one that allows the replies from the server in the other direction.

Feature	Description
Dynamic routing protocols	You can, however, add static routes for traffic originating on the ASA for bridge group member interfaces. You can also allow dynamic routing protocols through the ASA using an access rule.
Multicast IP routing	You can allow multicast traffic through the ASA by allowing it in an access rule.
QoS	—
VPN termination for through traffic	The transparent firewall supports site-to-site VPN tunnels for management connections only on bridge group member interfaces. It does not terminate VPN connections for traffic through the ASA. You can pass VPN traffic through the ASA using an access rule, but it does not terminate non-management connections. Clientless SSL VPN is also not supported.
Unified Communications	—

Passing Traffic For Routed-Mode Features

For features that are not directly supported on the transparent firewall, you can allow traffic to pass through so that upstream and downstream routers can support the functionality. For example, by using an access rule, you can allow DHCP traffic (instead of the unsupported DHCP relay feature) or multicast traffic such as that created by IP/TV. You can also establish routing protocol adjacencies through a transparent firewall; you can allow OSPF, RIP, EIGRP, or BGP traffic through based on an access rule. Likewise, protocols like HSRP or VRRP can pass through the ASA.

Default Settings

Default Mode

The default mode is routed mode.

Bridge Group Defaults

By default, all ARP packets are passed within the bridge group.

Guidelines for Firewall Mode

Context Mode Guidelines

Set the firewall mode per context.

Bridge Group Guidelines (Transparent Mode)

- You can create up to 250 bridge groups, with 4 interfaces per bridge group.
- Each directly-connected network must be on the same subnet.
- The ASA does not support traffic on secondary networks; only traffic on the same network as the BVI IP address is supported.
- An IP address for the BVI is required for each bridge group for to-the-device and from-the-device management traffic, as well as for data traffic to pass through the ASA. For IPv4 traffic, specify an IPv4 address. For IPv6 traffic, specify an IPv6 address.
- You can only configure IPv6 addresses manually.
- The BVI IP address must be on the same subnet as the connected network. You cannot set the subnet to a host subnet (255.255.255.255).
- Management interfaces are not supported as bridge group members.
- In transparent mode, you must use at least 1 bridge group; data interfaces must belong to a bridge group.
- In transparent mode, do not specify the BVI IP address as the default gateway for connected devices; devices need to specify the router on the other side of the ASA as the default gateway.
- In transparent mode, the *default* route, which is required to provide a return path for management traffic, is only applied to management traffic from one bridge group network. This is because the default route specifies an interface in the bridge group as well as the router IP address on the bridge group network, and you can only define one default route. If you have management traffic from more than one bridge group network, you need to specify a regular static route that identifies the network from which you expect management traffic.
- In transparent mode, PPPoE is not supported for the Management interface.
- Bidirectional Forwarding Detection (BFD) echo packets are not allowed through the ASA when using bridge group members. If there are two neighbors on either side of the ASA running BFD, then the ASA will drop BFD echo packets because they have the same source and destination IP address and appear to be part of a LAND attack.

Additional Guidelines and Limitations

- When you change firewall modes, the ASA clears the running configuration because many commands are not supported for both modes. The startup configuration remains unchanged. If you reload without saving, then the startup configuration is loaded, and the mode reverts back to the original setting. See Set the Firewall Mode, on page 157 for information about backing up your configuration file.
- If you download a text configuration to the ASA that changes the mode with the **firewall transparent** command, be sure to put the command at the top of the configuration; the ASA changes the mode as soon as it reads the command and then continues reading the configuration you downloaded. If the command appears later in the configuration, the ASA clears all the preceding lines in the configuration. See Set the ASA Image, ASDM, and Startup Configuration, on page 892 for information about downloading text files.

Set the Firewall Mode

This section describes how to change the firewall mode.

Note We recommend that you set the firewall mode before you perform any other configuration because changing the firewall mode clears the running configuration.

Before you begin

When you change modes, the ASA clears the running configuration (see Guidelines for Firewall Mode, on page 155 for more information).

- If you already have a populated configuration, be sure to back up your configuration before changing the mode; you can use this backup for reference when creating your new configuration. See Back Up and Restore Configurations or Other Files, on page 894.
- Use the CLI at the console port to change the mode. If you use any other type of session, including the ASDM Command Line Interface tool or SSH, you will be disconnected when the configuration is cleared, and you will have to reconnect to the ASA using the console port in any case.
- Set the mode within the context.

Note To set the firewall mode to transparent and also configure ASDM management access after the configuration is cleared, see Configure ASDM Access, on page 27.

Procedure

Set the firewall mode to transparent:

firewall transparent

Example:

```
ciscoasa(config)# firewall transparent
```

To change the mode to routed, enter the **no firewall transparent** command.

Note You are not prompted to confirm the firewall mode change; the change occurs immediately.

Examples for Firewall Mode

This section includes examples of how traffic moves through the ASA in the routed and transparent firewall mode.

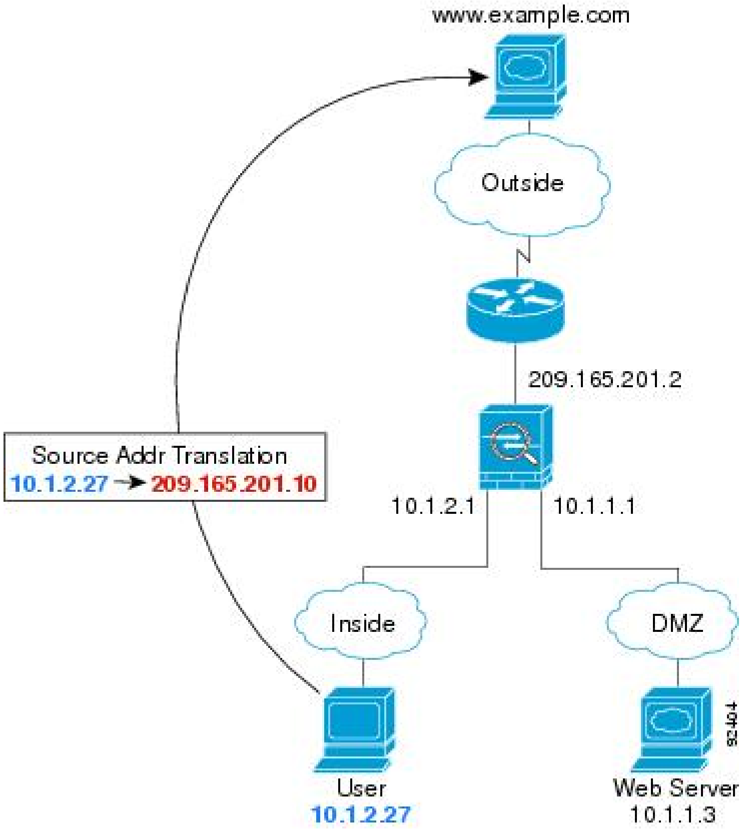
How Data Moves Through the ASA in Routed Firewall Mode

The following sections describe how data moves through the ASA in routed firewall mode in multiple scenarios.

An Inside User Visits a Web Server

The following figure shows an inside user accessing an outside web server.

Figure 16: Inside to Outside



The following steps describe how data moves through the ASA:

1. The user on the inside network requests a web page from www.example.com.
2. The ASA receives the packet and because it is a new session, it verifies that the packet is allowed according to the terms of the security policy.

   For multiple context mode, the ASA first classifies the packet to a context.

3. The ASA translates the real address (10.1.2.27) to the mapped address 209.165.201.10, which is on the outside interface subnet.

   The mapped address could be on any subnet, but routing is simplified when it is on the outside interface subnet.

4. The ASA then records that a session is established and forwards the packet from the outside interface.

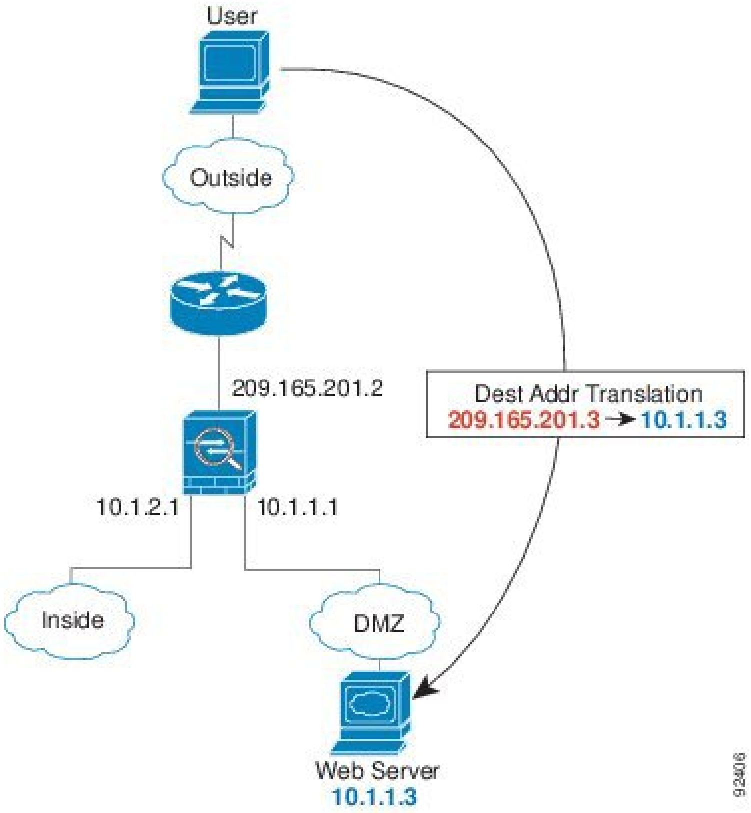
5. When www.example.com responds to the request, the packet goes through the ASA, and because the session is already established, the packet bypasses the many lookups associated with a new connection. The ASA performs NAT by untranslating the global destination address to the local user address, 10.1.2.27.

6. The ASA forwards the packet to the inside user.

An Outside User Visits a Web Server on the DMZ

The following figure shows an outside user accessing the DMZ web server.

Figure 17: Outside to DMZ



The following steps describe how data moves through the ASA:

1. A user on the outside network requests a web page from the DMZ web server using the mapped address of 209.165.201.3, which is on the outside interface subnet.

2. The ASA receives the packet and untranslates the mapped address to the real address 10.1.1.3.

3. Because it is a new session, the ASA verifies that the packet is allowed according to the terms of the security policy.

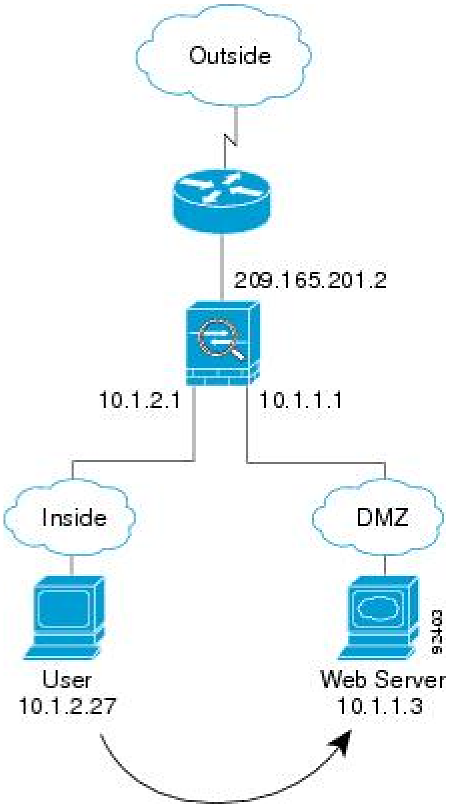
   For multiple context mode, the ASA first classifies the packet to a context.

4. The ASA then adds a session entry to the fast path and forwards the packet from the DMZ interface.
5. When the DMZ web server responds to the request, the packet goes through the ASA and because the session is already established, the packet bypasses the many lookups associated with a new connection. The ASA performs NAT by translating the real address to 209.165.201.3.
6. The ASA forwards the packet to the outside user.

An Inside User Visits a Web Server on the DMZ

The following figure shows an inside user accessing the DMZ web server.

Figure 18: Inside to DMZ



The following steps describe how data moves through the ASA:

1. A user on the inside network requests a web page from the DMZ web server using the destination address of 10.1.1.3.
2. The ASA receives the packet and because it is a new session, the ASA verifies that the packet is allowed according to the terms of the security policy.

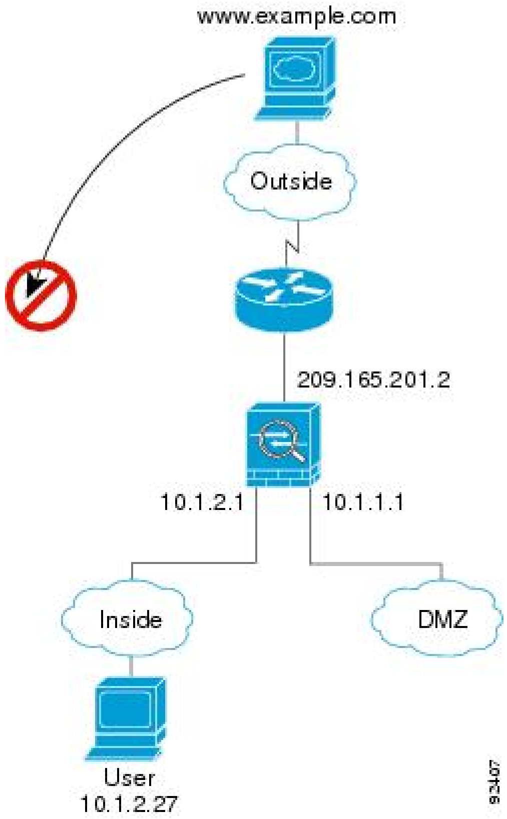
   For multiple context mode, the ASA first classifies the packet to a context.

3. The ASA then records that a session is established and forwards the packet out of the DMZ interface.
4. When the DMZ web server responds to the request, the packet goes through the fast path, which lets the packet bypass the many lookups associated with a new connection.
5. The ASA forwards the packet to the inside user.

An Outside User Attempts to Access an Inside Host

The following figure shows an outside user attempting to access the inside network.

Figure 19: Outside to Inside



The following steps describe how data moves through the ASA:

1. A user on the outside network attempts to reach an inside host (assuming the host has a routable IP address).

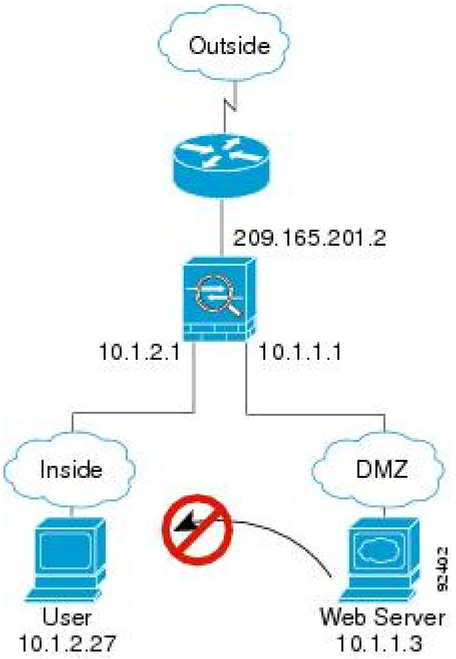
   If the inside network uses private addresses, no outside user can reach the inside network without NAT. The outside user might attempt to reach an inside user by using an existing NAT session.
2. The ASA receives the packet and because it is a new session, it verifies if the packet is allowed according to the security policy.
3. The packet is denied, and the ASA drops the packet and logs the connection attempt.

   If the outside user is attempting to attack the inside network, the ASA employs many technologies to determine if a packet is valid for an already established session.

A DMZ User Attempts to Access an Inside Host

The following figure shows a user in the DMZ attempting to access the inside network.

Figure 20: DMZ to Inside



The following steps describe how data moves through the ASA:

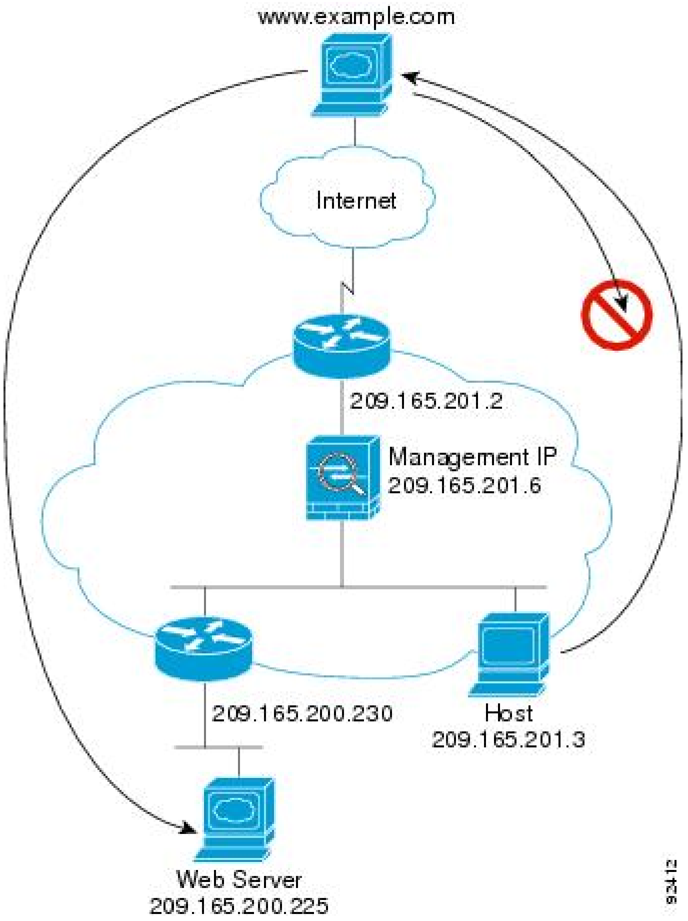
1. A user on the DMZ network attempts to reach an inside host. Because the DMZ does not have to route the traffic on the Internet, the private addressing scheme does not prevent routing.
2. The ASA receives the packet and because it is a new session, it verifies if the packet is allowed according to the security policy.

   The packet is denied, and the ASA drops the packet and logs the connection attempt.

How Data Moves Through the Transparent Firewall

The following figure shows a typical transparent firewall implementation with an inside network that contains a public web server. The ASA has an access rule so that the inside users can access Internet resources. Another access rule lets the outside users access only the web server on the inside network.

Figure 21: Typical Transparent Firewall Data Path

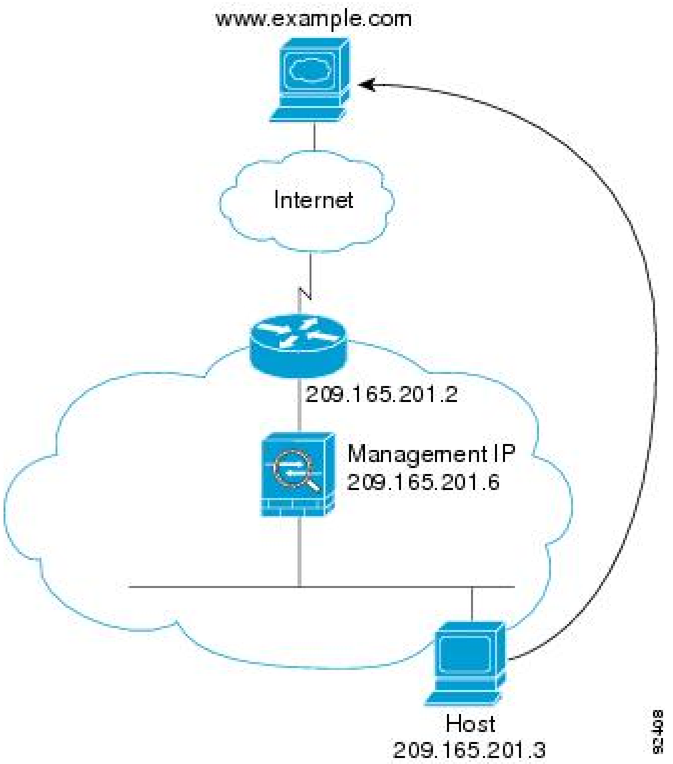


The following sections describe how data moves through the ASA.

An Inside User Visits a Web Server

The following figure shows an inside user accessing an outside web server.

Figure 22: Inside to Outside



The following steps describe how data moves through the ASA:

1. The user on the inside network requests a web page from www.example.com.
2. The ASA receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy.

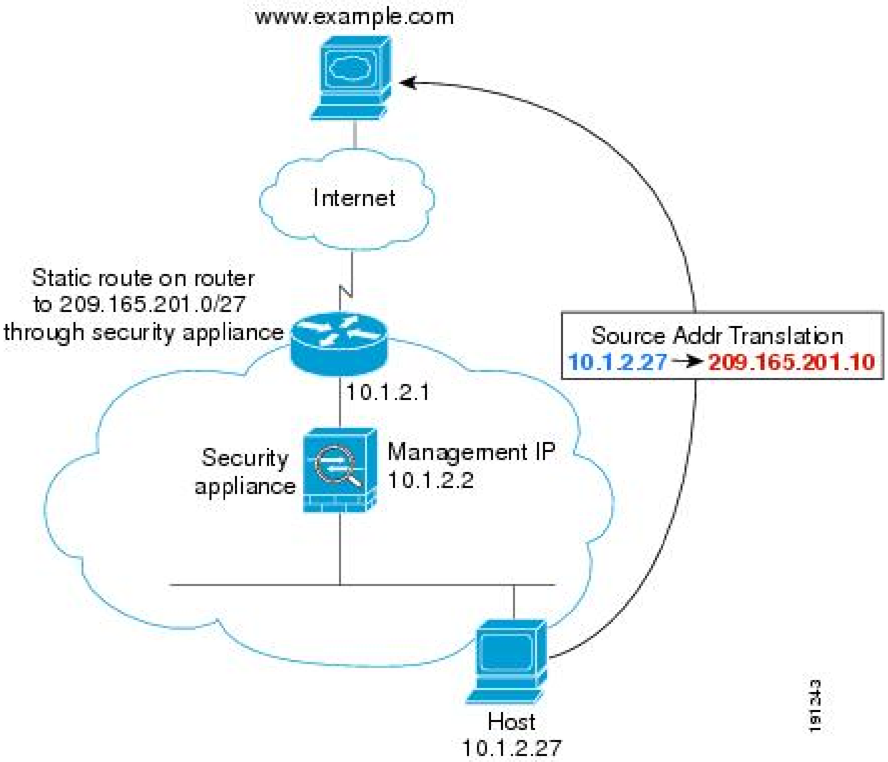
   For multiple context mode, the ASA first classifies the packet to a context.
3. The ASA records that a session is established.
4. If the destination MAC address is in its table, the ASA forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 209.165.201.2.

   If the destination MAC address is not in the ASA table, it attempts to discover the MAC address by sending an ARP request or a ping. The first packet is dropped.
5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The ASA forwards the packet to the inside user.

An Inside User Visits a Web Server Using NAT

The following figure shows an inside user accessing an outside web server.

Figure 23: Inside to Outside with NAT



The following steps describe how data moves through the ASA:

1. The user on the inside network requests a web page from www.example.com.

2. The ASA receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy.

   For multiple context mode, the ASA first classifies the packet according to a unique interface.

3. The ASA translates the real address (10.1.2.27) to the mapped address 209.165.201.10.

   Because the mapped address is not on the same network as the outside interface, then be sure the upstream router has a static route to the mapped network that points to the ASA.

4. The ASA then records that a session is established and forwards the packet from the outside interface.

5. If the destination MAC address is in its table, the ASA forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 10.1.2.1.

   If the destination MAC address is not in the ASA table, then it attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.

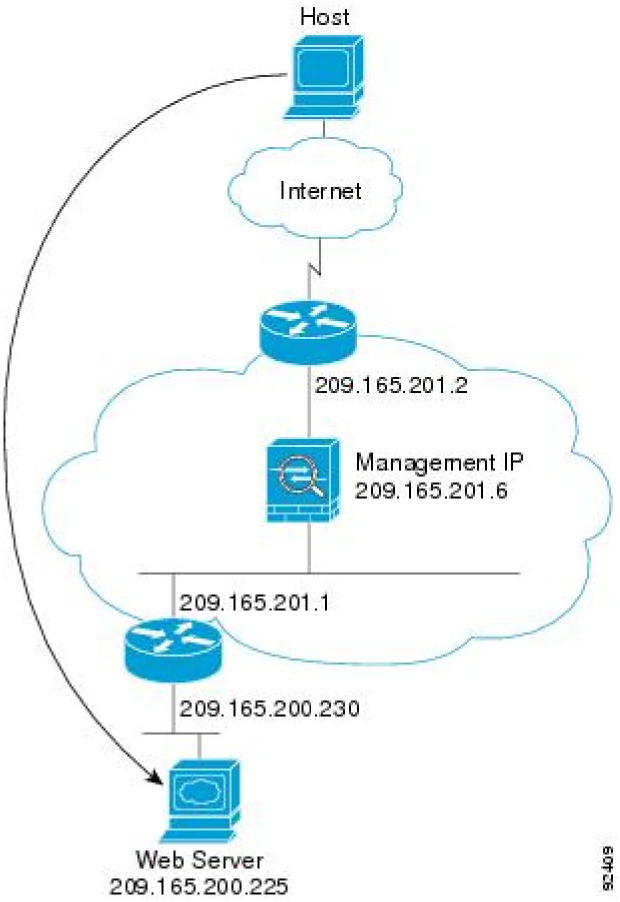
6. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.

7. The ASA performs NAT by untranslating the mapped address to the real address, 10.1.2.27.

An Outside User Visits a Web Server on the Inside Network

The following figure shows an outside user accessing the inside web server.

Figure 24: Outside to Inside



The following steps describe how data moves through the ASA:

1. A user on the outside network requests a web page from the inside web server.
2. The ASA receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy.

   For multiple context mode, the ASA first classifies the packet to a context.
3. The ASA records that a session is established.
4. If the destination MAC address is in its table, the ASA forwards the packet out of the inside interface. The destination MAC address is that of the downstream router, 209.165.201.1.

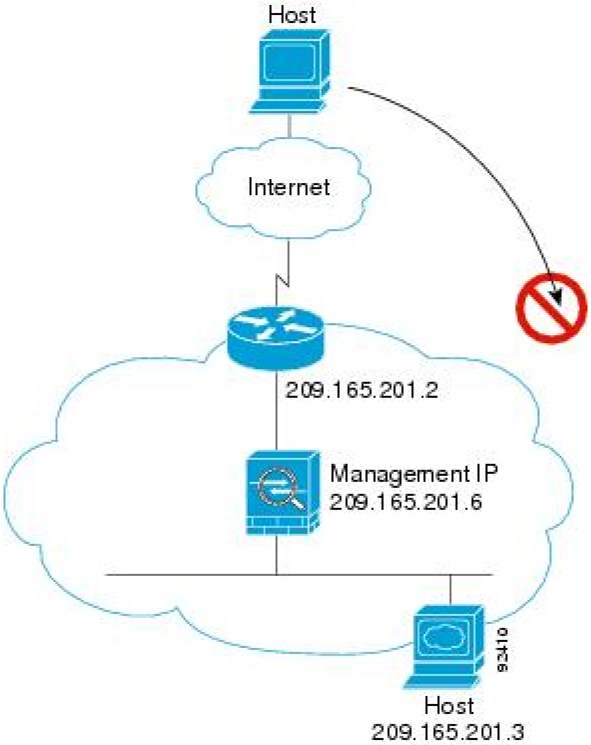
   If the destination MAC address is not in the ASA table, then it attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.
5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.

6. The ASA forwards the packet to the outside user.

An Outside User Attempts to Access an Inside Host

The following figure shows an outside user attempting to access a host on the inside network.

Figure 25: Outside to Inside



The following steps describe how data moves through the ASA:

1. A user on the outside network attempts to reach an inside host.
2. The ASA receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies if the packet is allowed according to the terms of the security policy.

   For multiple context mode, the ASA first classifies the packet to a context.
3. The packet is denied because there is no access rule permitting the outside host, and the ASA drops the packet.
4. If the outside user is attempting to attack the inside network, the ASA employs many technologies to determine if a packet is valid for an already established session.

History for the Firewall Mode

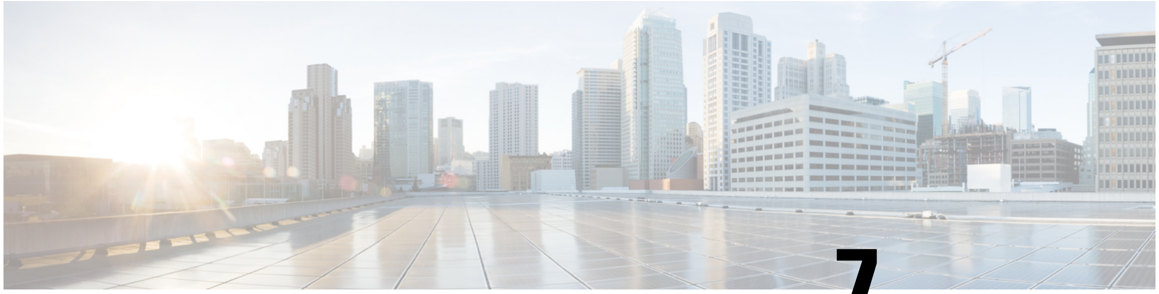
Table 4: Feature History for Firewall Mode

Feature Name	Platform Releases	Feature Information
Transparent Firewall Mode	7.0(1)	A transparent firewall is a Layer 2 firewall that acts like a “bump in the wire,” or a “stealth firewall,” and is not seen as a router hop to connected devices. We introduced the following commands: **firewall transparent**, **show firewall**.
Transparent firewall bridge groups	8.4(1)	If you do not want the overhead of security contexts, or want to maximize your use of security contexts, you can group interfaces together in a bridge group, and then configure multiple bridge groups, one for each network. Bridge group traffic is isolated from other bridge groups. You can configure up to 8 bridge groups in single mode or per context in multiple mode, with 4 interfaces maximum per bridge group. **Note** Although you can configure multiple bridge groups on the ASA 5505, the restriction of 2 data interfaces in transparent mode on the ASA 5505 means you can only effectively use 1 bridge group. We introduced the following commands: **interface bvi**, **bridge-group**, **show bridge-group**.
Mixed firewall mode support in multiple context mode	8.5(1)/9.0(1)	You can set the firewall mode independently for each security context in multiple context mode, so some can run in transparent mode while others run in routed mode. We modified the following command: **firewall transparent**.

Feature Name	Platform Releases	Feature Information
Transparent mode bridge group maximum increased to 250	9.3(1)	The bridge group maximum was increased from 8 to 250 bridge groups. You can configure up to 250 bridge groups in single mode or per context in multiple mode, with 4 interfaces maximum per bridge group. We modified the following commands: **interface bvi**, **bridge-group**.

PART II

High Availability and Scalability



CHAPTER 7

Multiple Context Mode

This chapter describes how to configure multiple security contexts on the Cisco ASA.

About Security Contexts

You can partition a single ASA into multiple virtual devices, known as security contexts. Each context acts as an independent device, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. For unsupported features in multiple context mode, see Guidelines for Multiple Context Mode, on page 184.

This section provides an overview of security contexts.

Common Uses for Security Contexts

You might want to use multiple security contexts in the following situations:

- You are a service provider and want to sell security services to many customers. By enabling multiple security contexts on the ASA, you can implement a cost-effective, space-saving solution that keeps all customer traffic separate and secure, and also eases configuration.

- You are a large enterprise or a college campus and want to keep departments completely separate.

- You are an enterprise that wants to provide distinct security policies to different departments.

- You have any network that requires more than one ASA.

Context Configuration Files

This section describes how the ASA implements multiple context mode configurations.

Context Configurations

For each context, the ASA includes a configuration that identifies the security policy, interfaces, and all the options you can configure on a standalone device. You can store context configurations in flash memory, or you can download them from a TFTP, FTP, or HTTP(S) server.

System Configuration

The system administrator adds and manages contexts by configuring each context configuration location, allocated interfaces, and other context operating parameters in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the ASA. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the *admin context*. The system configuration does include a specialized failover interface for failover traffic only.

Admin Context Configuration

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users. The admin context must reside on flash memory, and not remotely.

If your system is already in multiple context mode, or if you convert from single mode, the admin context is created automatically as a file on the internal flash memory called admin.cfg. This context is named “admin.” If you do not want to use admin.cfg as the admin context, you can change the admin context.

How the ASA Classifies Packets

Each packet that enters the ASA must be classified, so that the ASA can determine to which context to send a packet.

Note If the destination MAC address is a multicast or broadcast MAC address, the packet is duplicated and delivered to each context.

Valid Classifier Criteria

This section describes the criteria used by the classifier.

Note For management traffic destined for an interface, the interface IP address is used for classification.

The routing table is not used for packet classification.

Unique Interfaces

If only one context is associated with the ingress interface, the ASA classifies the packet into that context. In transparent firewall mode, unique interfaces for contexts are required, so this method is used to classify packets at all times.

Unique MAC Addresses

If multiple contexts share an interface, then the classifier uses unique MAC addresses assigned to the interface in each context. An upstream router cannot route directly to a context without unique MAC addresses. You can enable auto-generation of MAC addresses. You can also set the MAC addresses manually when you configure each interface.

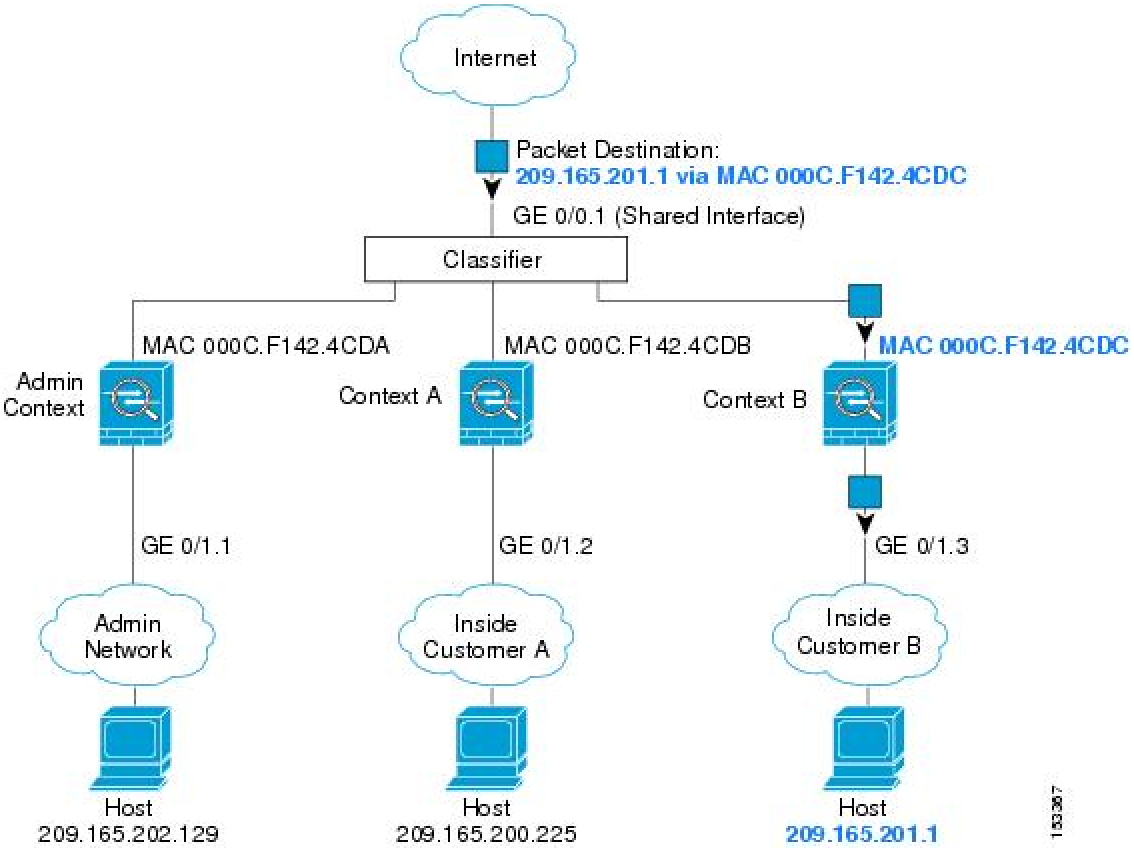
NAT Configuration

If you do not enable use of unique MAC addresses, then the ASA uses the mapped addresses in your NAT configuration to classify packets. We recommend using MAC addresses instead of NAT, so that traffic classification can occur regardless of the completeness of the NAT configuration.

Classification Examples

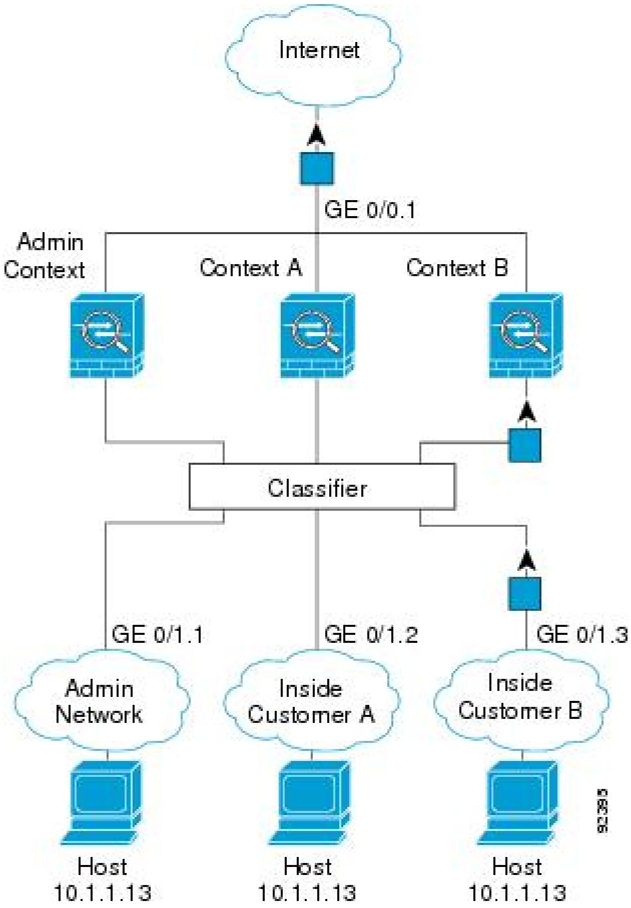
The following figure shows multiple contexts sharing an outside interface. The classifier assigns the packet to Context B because Context B includes the MAC address to which the router sends the packet.

Figure 26: Packet Classification with a Shared Interface Using MAC Addresses



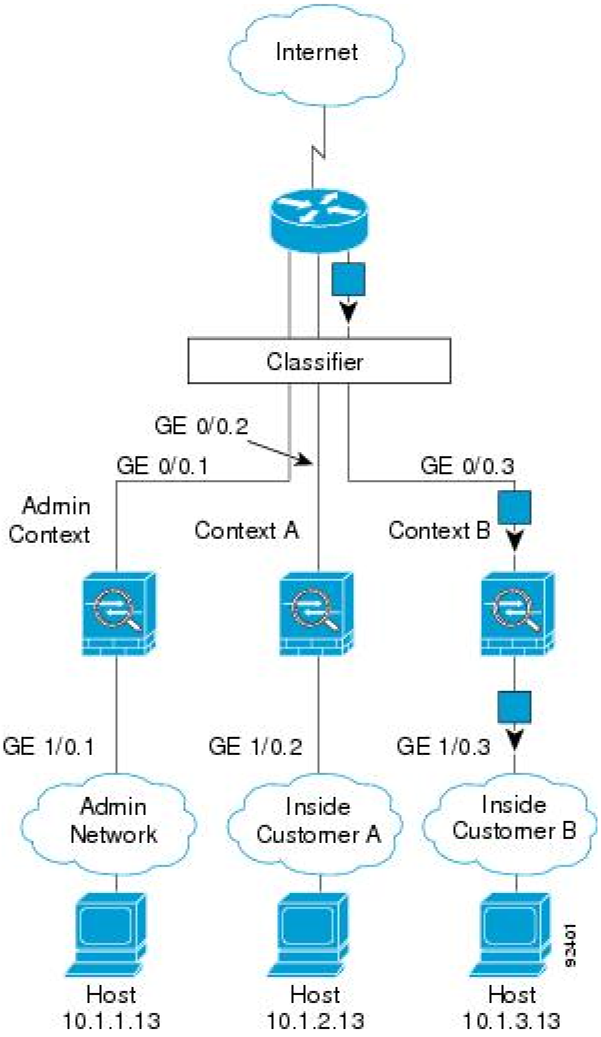
Note that all new incoming traffic must be classified, even from inside networks. The following figure shows a host on the Context B inside network accessing the Internet. The classifier assigns the packet to Context B because the ingress interface is Gigabit Ethernet 0/1.3, which is assigned to Context B.

Figure 27: Incoming Traffic from Inside Networks



For transparent firewalls, you must use unique interfaces. The following figure shows a packet destined to a host on the Context B inside network from the Internet. The classifier assigns the packet to Context B because the ingress interface is Gigabit Ethernet 1/0.3, which is assigned to Context B.

Figure 28: Transparent Firewall Contexts



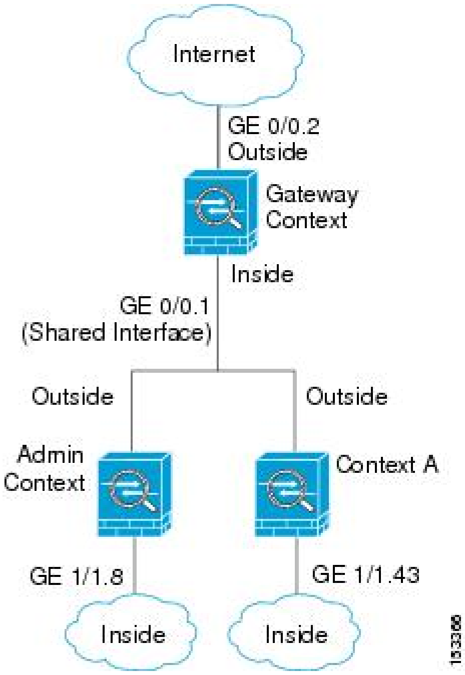
Cascading Security Contexts

Placing a context directly in front of another context is called *cascading contexts*; the outside interface of one context is the same interface as the inside interface of another context. You might want to cascade contexts if you want to simplify the configuration of some contexts by configuring shared parameters in the top context.

Note Cascading contexts requires unique MAC addresses for each context interface. Because of the limitations of classifying packets on shared interfaces without MAC addresses, we do not recommend using cascading contexts without unique MAC addresses.

The following figure shows a gateway context with two contexts behind the gateway.

Figure 29: Cascading Contexts



Management Access to Security Contexts

The ASA provides system administrator access in multiple context mode as well as access for individual context administrators.

System Administrator Access

You can access the ASA as a system administrator in two ways:

- Access the ASA console.

  From the console, you access the *system execution space*, which means that any commands you enter affect only the system configuration or the running of the system (for run-time commands).

- Access the admin context using Telnet, SSH, or ASDM.

As the system administrator, you can access all contexts.

The system execution space does not support any AAA commands, but you can configure its own enable password, as well as usernames in the local database to provide individual logins.

Context Administrator Access

You can access a context using Telnet, SSH, or ASDM. If you log in to a non-admin context, you can only access the configuration for that context. You can provide individual logins to the context.

Management Interface Usage

The Management interface is a separate interface just for management traffic.

In routed firewall mode, you can share the Management interface across all contexts.

In transparent firewall mode, the Management interface is special. In addition to the maximum allowed through-traffic interfaces, you can also use the Management interface as a separate management-only interface. However, in multiple context mode, you cannot share any interfaces across transparent contexts. You can instead use subinterfaces of the Management interface, and assign one to each context. However, only Firepower models and the ASA 5585-X allow subinterfaces on the Management interface. For ASA models other than the ASA 5585-X, you must use a data interface or a subinterface of a data interface, and add it to a bridge group within the context.

For the Firepower 9300 chassis transparent context, neither the Management interface nor subinterface retains its special status. In this case, you must treat it as a data interface, and add it to a bridge group. (Note that in single context mode, the Management interface does retain its special status.)

Another consideration about transparent mode: when you enable multiple context mode, all configured interfaces are automatically assigned to the Admin context. For example, if your default configuration includes the Management interface, then that interface will be assigned to the Admin context. One option is to leave the main interface allocated to the Admin context and manage it using the native VLAN, and then use subinterfaces to manage each context. Keep in mind that if you make the Admin context transparent, its IP address will be removed; you have to assign it to a bridge group and assign the IP address to the BVI.

About Resource Management

By default, all security contexts have unlimited access to the resources of the ASA, except where maximum limits per context are enforced; the only exception is VPN resources, which are disabled by default. If you find that one or more contexts use too many resources, and they cause other contexts to be denied connections, for example, then you can configure resource management to limit the use of resources per context. For VPN resources, you must configure resource management to allow any VPN tunnels.

Resource Classes

The ASA manages resources by assigning contexts to resource classes. Each context uses the resource limits set by the class. To use the settings of a class, assign the context to the class when you define the context. All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to default. You can only assign a context to one resource class. The exception to this rule is that limits that are undefined in the member class are inherited from the default class; so in effect, a context could be a member of default plus another class.

Resource Limits

You can set the limit for individual resources as a percentage (if there is a hard system limit) or as an absolute value.

For most resources, the ASA does not set aside a portion of the resources for each context assigned to the class; rather, the ASA sets the maximum limit for a context. If you oversubscribe resources, or allow some resources to be unlimited, a few contexts can “use up” those resources, potentially affecting service to other contexts. The exception is VPN resource types, which you cannot oversubscribe, so the resources assigned to each context are guaranteed. To accommodate temporary bursts of VPN sessions beyond the amount assigned, the ASA supports a “burst” VPN resource type, which is equal to the remaining unassigned VPN sessions. The burst sessions *can* be oversubscribed, and are available to contexts on a first-come, first-served basis.

Default Class

All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to the default class.

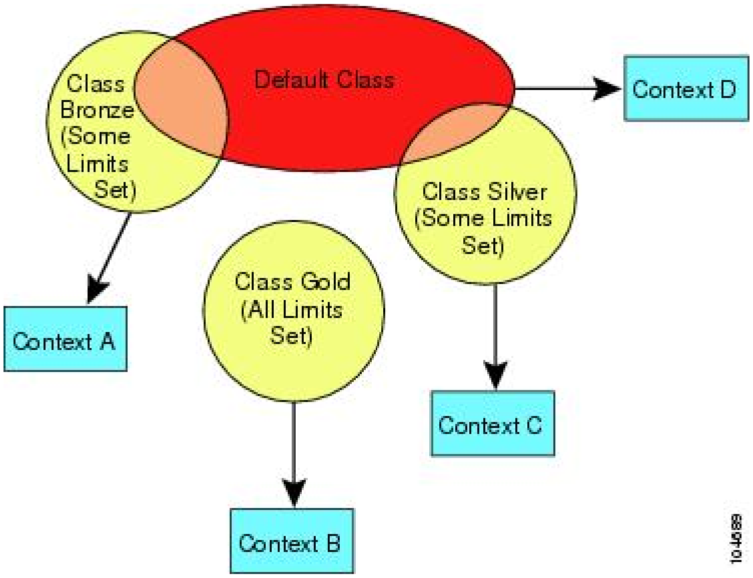
If a context belongs to a class other than the default class, those class settings always override the default class settings. However, if the other class has any settings that are not defined, then the member context uses the default class for those limits. For example, if you create a class with a 2 percent limit for all concurrent connections, but no other limits, then all other limits are inherited from the default class. Conversely, if you create a class with a limit for all resources, the class uses no settings from the default class.

For most resources, the default class provides unlimited access to resources for all contexts, except for the following limits:

- Telnet sessions—5 sessions. (The maximum per context.)
- SSH sessions—5 sessions. (The maximum per context.)
- ASDM sessions—32 sessions. (The maximum per context.)
- IPsec sessions—5 sessions. (The maximum per context.)
- MAC addresses—65,535 entries. (The maximum for the system.)
- VPN site-to-site tunnels—0 sessions. (You must manually configure the class to allow any VPN sessions.)

The following figure shows the relationship between the default class and other classes. Contexts A and C belong to classes with some limits set; other limits are inherited from the default class. Context B inherits no limits from default because all limits are set in its class, the Gold class. Context D was not assigned to a class, and is by default a member of the default class.

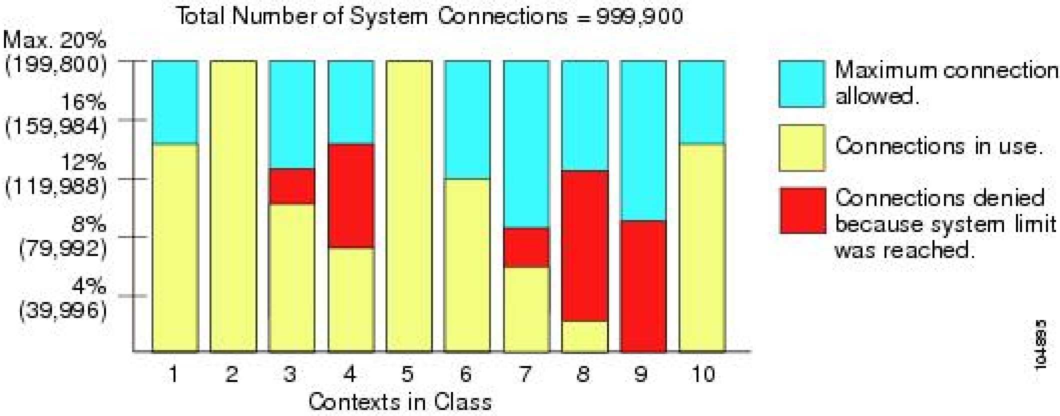
Figure 30: Resource Classes


Class Bronze (Some Limits Set)
Default Class
Context D
Class Silver (Some Limits Set)
Class Gold (All Limits Set)
Context A
Context C
Context B
104689

Use Oversubscribed Resources

You can oversubscribe the ASA by assigning more than 100 percent of a resource across all contexts (with the exception of non-burst VPN resources). For example, you can set the Bronze class to limit connections to 20 percent per context, and then assign 10 contexts to the class for a total of 200 percent. If contexts concurrently use more than the system limit, then each context gets less than the 20 percent you intended.

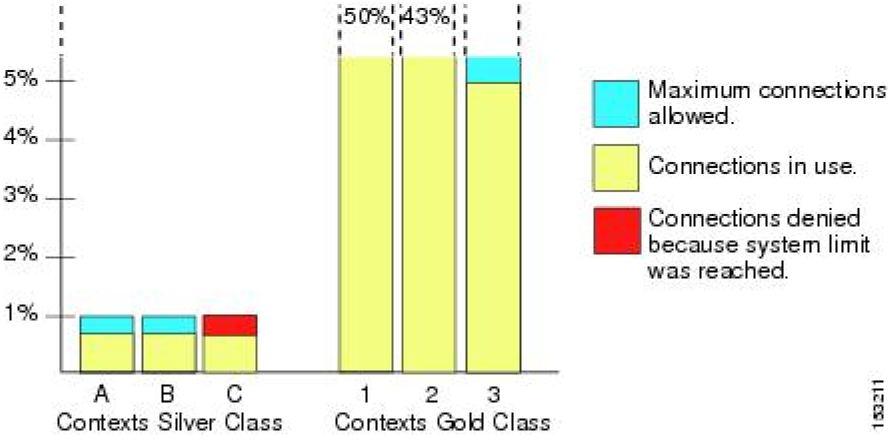
Figure 31: Resource Oversubscription



Use Unlimited Resources

The ASA lets you assign unlimited access to one or more resources in a class, instead of a percentage or absolute number. When a resource is unlimited, contexts can use as much of the resource as the system has available. For example, Context A, B, and C are in the Silver Class, which limits each class member to 1 percent of the connections, for a total of 3 percent; but the three contexts are currently only using 2 percent combined. Gold Class has unlimited access to connections. The contexts in the Gold Class can use more than the 97 percent of “unassigned” connections; they can also use the 1 percent of connections not currently in use by Context A, B, and C, even if that means that Context A, B, and C are unable to reach their 3 percent combined limit. Setting unlimited access is similar to oversubscribing the ASA, except that you have less control over how much you oversubscribe the system.

Figure 32: Unlimited Resources



About MAC Addresses

You can manually assign MAC addresses to override the default. For multiple context mode, you can automatically generate unique MAC addresses (for all interfaces assigned to a context).

Note You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

MAC Addresses in Multiple Context Mode

The MAC address is used to classify packets within a context. If you share an interface, but do not have unique MAC addresses for the interface in each context, then other classification methods are attempted that might not provide full coverage.

To allow contexts to share interfaces, you should enable auto-generation of virtual MAC addresses to each shared context interface. On the ASASM only, auto-generation is enabled by default in multiple context mode.

Automatic MAC Addresses

In multiple context mode, auto-generation assigns unique MAC addresses to all interfaces assigned to a context.

If you manually assign a MAC address and also enable auto-generation, then the manually assigned MAC address is used. If you later remove the manual MAC address, the auto-generated address is used, if enabled.

In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface.

Because auto-generated addresses (when using a prefix) start with A2, you cannot start manual MAC addresses with A2 if you also want to use auto-generation.

The ASA generates the MAC address using the following format:

A2*xx.yyzz.zzzz*

Where *xx.yy* is a user-defined prefix or an autogenerated prefix based on the last two bytes of the interface MAC address, and *zz.zzzz* is an internal counter generated by the ASA. For the standby MAC address, the address is identical except that the internal counter is increased by 1.

For an example of how the prefix is used, if you set a prefix of 77, then the ASA converts 77 into the hexadecimal value 004D (*yyxx*). When used in the MAC address, the prefix is reversed (*xxyy*) to match the ASA native form:

A2**4D.00***zz.zzzz*

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03***zz.zzzz*

Note The MAC address format without a prefix is a legacy version. See the **mac-address auto** command in the command reference for more information about the legacy format.

VPN Support

For VPN resources, you must configure resource management to allow any VPN tunnels.

You can use site-to-site VPN in multiple context mode.

Remote access VPN is not supported.

Licensing for Multiple Context Mode

Model	**License Requirement**
ASA 5506-X	No support.
ASA 5508-X	Security Plus License: 2 contexts. *Optional license: 5 contexts.*
ASA 5512-X	• Base License: No support. • Security Plus License: 2 contexts. *Optional license: 5 contexts.*
ASA 5515-X	Base License: 2 contexts. *Optional license: 5 contexts.*
ASA 5516-X	Security Plus License: 2 contexts. *Optional license: 5 contexts.*
ASA 5525-X	Base License: 2 contexts. *Optional licenses: 5, 10, or 20 contexts.*
ASA 5545-X	Base License: 2 contexts. *Optional licenses: 5, 10, 20, or 50 contexts.*
ASA 5555-X	Base License: 2 contexts. *Optional licenses: 5, 10, 20, 50, or 100 contexts.*
ASA 5585-X with SSP-10	Base License: 2 contexts. *Optional licenses: 5, 10, 20, 50, or 100 contexts.*
ASA 5585-X with SSP-20, -40, and -60	Base License: 2 contexts. *Optional licenses: 5, 10, 20, 50, 100, or 250 contexts.*
ASASM	Base License: 2 contexts. *Optional licenses: 5, 10, 20, 50, 100, or 250 contexts.*

Model	License Requirement
Firepower 9300	Base License: 10 contexts. *Optional licenses: up to 250 contexts, in increments of 10.*
ASAv	No support.

Note If the Admin context only contains management-only interfaces, and does not include any data interfaces for through traffic, then it does not count against the limit.

Prerequisites for Multiple Context Mode

After you are in multiple context mode, connect to the system or the admin context to access the system configuration. You cannot configure the system from a non-admin context. By default, after you enable multiple context mode, you can connect to the admin context by using the default management IP address.

Guidelines for Multiple Context Mode

Failover

Active/Active mode failover is only supported in multiple context mode.

IPv6

Cross-context IPv6 routing is not supported.

Unsupported Features

Multiple context mode does not support the following features:

- RIP
- OSPFv3. (OSPFv2 is supported.)
- Multicast routing
- Threat Detection
- Unified Communications
- QoS
- Remote access VPN. (Site-to-site VPN is supported.)
- Static route tracking

Additional Guidelines

- The context mode (single or multiple) is not stored in the configuration file, even though it does endure reboots. If you need to copy your configuration to another device, set the mode on the new device to match.
- If you store context configurations in the root directory of flash memory, on some models you might run out of room in that directory, even though there is available memory. In this case, create a subdirectory for your configuration files. Background: some models, such as the ASA 5585-X, use the FAT 16 file system for internal flash memory, and if you do not use 8.3-compliant short names, or use uppercase characters, then fewer than 512 files and folders can be stored because the file system uses up slots to store long file names (see http://support.microsoft.com/kb/120138/en-us).

Defaults for Multiple Context Mode

- By default, the ASA is in single context mode.
- See Default Class, on page 180.

Configure Multiple Contexts

Procedure

Step 1 Enable or Disable Multiple Context Mode, on page 185.

Step 2 (Optional) Configure a Class for Resource Management, on page 187.

Note For VPN support, you must configure VPN resources in a resource class; the default class does not allow VPN.

Step 3 Configure interfaces in the system execution space.

- ASA 5500-X—Basic Interface Configuration, on page 401.
- Firepower 9300—Logical Devices for the Firepower 9300, on page 131
- ASASM—ASASM quick start guide.

Step 4 Configure a Security Context, on page 191.

Step 5 (Optional) Assign MAC Addresses to Context Interfaces Automatically, on page 194.

Step 6 Complete interface configuration in the context. See Routed and Transparent Mode Interfaces, on page 451.

Enable or Disable Multiple Context Mode

Your ASA might already be configured for multiple security contexts depending on how you ordered it from Cisco. If you need to convert from single mode to multiple mode, follow the procedures in this section.

Enable Multiple Context Mode

When you convert from single mode to multiple mode, the ASA converts the running configuration into two files: a new startup configuration that comprises the system configuration, and admin.cfg that comprises the admin context (in the root directory of the internal flash memory). The original running configuration is saved as old_running.cfg (in the root directory of the internal flash memory). The original startup configuration is not saved. The ASA automatically adds an entry for the admin context to the system configuration with the name “admin.”

Before you begin

Back up your startup configuration if it differs from the running configuration. When you convert from single mode to multiple mode, the ASA converts the running configuration into two files. The original startup configuration is not saved. See Back Up and Restore Configurations or Other Files, on page 894.

Procedure

Change to multiple context mode.

mode multiple

Example:

You are prompted to change the mode and convert the configuration, and then the system reloads.

Note You will have to regenerate the RSA key pair in the Admin context before you can reestablish an SSH connection. From the console, enter the **crypto key generate rsa modulus** command. See Configure SSH Access, on page 837 for more information.

Example:

```
ciscoasa(config)# mode multiple
WARNING: This command will change the behavior of the device
WARNING: This command will initiate a Reboot
Proceed with change mode? [confirm]
Convert the system configuration? [confirm]
!
The old running configuration file will be written to flash

Converting the configuration - this may take several minutes for a large configuration

The admin context configuration will be written to flash

The new running configuration file was written to flash
Security context mode: multiple
ciscoasa(config)#


***
*** --- START GRACEFUL SHUTDOWN ---
***
*** Message to all terminals:
***
***   change mode
Shutting down isakmp
Shutting down webvpn
Shutting down License Controller
Shutting down File system
```

```
***
*** --- SHUTDOWN NOW ---
***
*** Message to all terminals:
***
***   change mode
```

Restore Single Context Mode

To copy the old running configuration to the startup configuration and to change the mode to single mode, perform the following steps.

Before you begin

Perform this procedure in the system execution space.

Procedure

Step 1 Copy the backup version of your original running configuration to the current startup configuration:

copy disk0:old_running.cfg startup-config

Example:

```
ciscoasa(config)# copy disk0:old_running.cfg startup-config
```

Step 2 Set the mode to single mode:

mode single

Example:

```
ciscoasa(config)# mode single
```

You are prompted to reboot the ASA.

Configure a Class for Resource Management

To configure a class in the system configuration, perform the following steps. You can change the value of a particular resource limit by reentering the command with a new value.

Before you begin

- Perform this procedure in the system execution space.
- The following table lists the resource types and the limits. See also the **show resource types** command.

Note If the System Limit is N/A, then you cannot set a percentage of the resource because there is no hard system limit for the resource.

Table 5: Resource Names and Limits

Resource Name	Rate or Concurrent	Minimum and Maximum Number per Context	System Limit	Description
asdm	Concurrent	1 minimum 32 maximum	200	ASDM management sessions. ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the system limit of 200 ASDM sessions represents a limit of 400 HTTPS sessions.
conns	Concurrent or Rate	N/A	Concurrent connections: See Supported Feature Licenses Per Model, on page 73 for the connection limit available for your model. Rate: N/A	TCP or UDP connections between any two hosts, including connections between one host and multiple other hosts. **Note** Syslog messages are generated for whichever limit is lower, xlates or conns. For example, if you set the xlates limit to 7 and the conns to 9, then the ASA only generates syslog message 321001 (“Resource 'xlates' limit of 7 reached for context 'ctx1'”) and not 321002 (“Resource 'conn rate' limit of 5 reached for context 'ctx1'”).
hosts	Concurrent	N/A	N/A	Hosts that can connect through the ASA.
http	Concurrent	1 minimum 6 maximum	100	Non-ASDM HTTPS sessions
inspects	Rate	N/A	N/A	Application inspections per second.
mac-addresses	Concurrent	N/A	65,535	For transparent firewall mode, the number of MAC addresses allowed in the MAC address table.

Resource Name	Rate or Concurrent	Minimum and Maximum Number per Context	System Limit	Description
routes	Concurrent	N/A	N/A	Dynamic routes.
vpn burst other	Concurrent	N/A	The Other VPN session amount for your model minus the sum of the sessions assigned to all contexts for **vpn other**.	The number of site-to-site VPN sessions allowed beyond the amount assigned to a context with **vpn other**. For example, if your model supports 5000 sessions, and you assign 4000 sessions across all contexts with **vpn other**, then the remaining 1000 sessions are available for **vpn burst other**. Unlike **vpn other**, which guarantees the sessions to the context, **vpn burst other** can be oversubscribed; the burst pool is available to all contexts on a first-come, first-served basis.
vpn other	Concurrent	N/A	See Supported Feature Licenses Per Model, on page 73 for the Other VPN sessions available for your model.	Site-to-site VPN sessions. You cannot oversubscribe this resource; all context assignments combined cannot exceed the model limit. The sessions you assign for this resource are guaranteed to the context.
ikev1 in-negotiation	Concurrent (percentage only)	N/A	A percentage of the Other VPN sessions assigned to this context. See the **vpn other** resources to assign sessions to the context.	Incoming IKEv1 SA negotiations, as a percentage of the context Other VPN limit.
ssh	Concurrent	1 minimum 5 maximum	100	SSH sessions.
syslogs	Rate	N/A	N/A	Syslog messages per second.
telnet	Concurrent	1 minimum 5 maximum	100	Telnet sessions.
xlates	Concurrent	N/A	N/A	Network address translations.

Procedure

Step 1 Specify the class name and enter the class configuration mode:

class *name*

Example:

```
ciscoasa(config)# class gold
```

The *name* is a string up to 20 characters long. To set the limits for the default class, enter **default** for the name.

Step 2 Set the resource limit for a resource type:

limit-resource [**rate**] *resource_name number*[**%**]

Example:

```
ciscoasa(config-class)# limit-resource rate inspects 10
```

- See the preceding table for a list of resource types. If you specify **all**, then all resources are configured with the same value. If you also specify a value for a particular resource, the limit overrides the limit set for **all**.
- Enter the **rate** argument to set the rate per second for certain resources.
- For most resources, specify **0** for the *number* to set the resource to be unlimited or to be the system limit, if available. For VPN resources, **0** sets the limit to none.
- For resources that do not have a system limit, you cannot set the percentage (**%**); you can only set an absolute value.

Example

For example, to set the default class limit for conns to 10 percent instead of unlimited, and to allow 5 site-to-site VPN tunnels with 2 tunnels allowed for VPN burst, enter the following commands:

```
ciscoasa(config)# class default
ciscoasa(config-class)# limit-resource conns 10%
ciscoasa(config-class)# limit-resource vpn other 5
ciscoasa(config-class)# limit-resource vpn burst other 2
```

All other resources remain at unlimited.

To add a class called gold, enter the following commands:

```
ciscoasa(config)# class gold
ciscoasa(config-class)# limit-resource mac-addresses 10000
ciscoasa(config-class)# limit-resource conns 15%
ciscoasa(config-class)# limit-resource rate conns 1000
ciscoasa(config-class)# limit-resource rate inspects 500
ciscoasa(config-class)# limit-resource hosts 9000
ciscoasa(config-class)# limit-resource asdm 5
ciscoasa(config-class)# limit-resource ssh 5
ciscoasa(config-class)# limit-resource rate syslogs 5000
ciscoasa(config-class)# limit-resource telnet 5
ciscoasa(config-class)# limit-resource xlates 36000
ciscoasa(config-class)# limit-resource routes 5000
ciscoasa(config-class)# limit-resource vpn other 10
ciscoasa(config-class)# limit-resource vpn burst other 5
```

Configure a Security Context

The security context definition in the system configuration identifies the context name, configuration file URL, interfaces that a context can use, and other settings.

Before you begin

- Perform this procedure in the system execution space.
- Configure interfaces. For transparent mode contexts, you cannot share interfaces between contexts, so you might want to use subinterfaces. To plan for Management interface usage, see Management Interface Usage, on page 178.
  - ASA 5500-X—Basic Interface Configuration, on page 401.
  - Firepower 9300—Logical Devices for the Firepower 9300, on page 131
  - ASASM—ASASM quick start guide.
- If you do not have an admin context (for example, if you clear the configuration) then you must first specify the admin context name by entering the following command:

  ```
  ciscoasa(config)# admin-context name
  ```

  Although this context does not exist yet in your configuration, you can subsequently enter the **context** *name* command to continue the admin context configuration.

Procedure

Step 1 Add or modify a context:

context *name*

Example:

```
ciscoasa(config)# context admin
```

The *name* is a string up to 32 characters long. This name is case sensitive, so you can have two contexts named “customerA” and “CustomerA,” for example. You can use letters, digits, or hyphens, but you cannot start or end the name with a hyphen.

Note “System” or “Null” (in upper or lower case letters) are reserved names, and cannot be used.

Step 2 (Optional) Add a description for this context:

description *text*

Example:

```
ciscoasa(config-ctx)# description Admin Context
```

Step 3 Specify the interfaces you can use in the context:

To allocate an interface:

allocate-interface *interface_id* [*mapped_name*] [**visible** | **invisible**]

To allocate one or more subinterfaces:

allocate-interface *interface_id***.***subinterface* [**-***interface_id***.***subinterface*] [*mapped_name*[**-***mapped_name*]] [**visible** | **invisible**]

Example:

```
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.100 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.200 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/2.300-gigabitethernet0/2.305
int3-int8
```

Note Do not include a space between the interface type and the port number.

- Enter these commands multiple times to specify different ranges. If you remove an allocation with the **no** form of this command, then any context commands that include this interface are removed from the running configuration.
- You can assign the same interfaces to multiple contexts in routed mode, if desired. Transparent mode does not allow shared interfaces.
- The *mapped_name* is an alphanumeric alias for the interface that can be used within the context instead of the interface ID. If you do not specify a mapped name, the interface ID is used within the context. For security purposes, you might not want the context administrator to know which interfaces the context is using. A mapped name must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, or an underscore. For example, you can use the following names: **int0**, **inta**, **int_0**.
- If you specify a range of subinterfaces, you can specify a matching range of mapped names. Follow these guidelines for ranges:
  - The mapped name must consist of an alphabetic portion followed by a numeric portion. The alphabetic portion of the mapped name must match for both ends of the range. For example, enter the following range: **int0-int10**. If you enter gig0/1.1-gig0/1.5 **happy**1-**sad**5, for example, the command fails.
  - The numeric portion of the mapped name must include the same quantity of numbers as the subinterface range. For example, both ranges include 100 interfaces:**gigabitethernet0/0.100-gigabitethernet0/0.199 int1-int100**. If you enter gig0/0.**100**-gig0/0.**199** int**1**-int**15**, for example, the command fails.
- Specify **visible** to see the real interface ID in the **show interface** command if you set a mapped name. The default **invisible** keyword shows only the mapped name.

Step 4 Identify the URL from which the system downloads the context configuration:

config-url *url*

Example:

```
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
```

Step 5 (Optional) Assign the context to a resource class:

member *class_name*

Example:

```
ciscoasa(config-ctx)# member gold
```

If you do not specify a class, the context belongs to the default class. You can only assign a context to one resource class.

Step 6 (Optional) Assign an IPS virtual sensor to this context if you have the IPS module installed:

allocate-ips *sensor_name* [*mapped_name*] [**default**]

Example:

```
ciscoasa(config-ctx)# allocate-ips sensor1 highsec
```

See the IPS quick start guide for detailed information about virtual sensors.

- When you add a context URL, the system immediately loads the context so that it is running, if the configuration is available.
- Enter the allocate-interface commands before you enter the **config-url** command. If you enter the **config-url** command first, the ASA loads the context configuration immediately. If the context contains any commands that refer to (not yet configured) interfaces, those commands fail.
- The filename does not require a file extension, although we recommend using ".cfg". The server must be accessible from the admin context. If the configuration file is not available, you see the following warning message:

```
WARNING: Could not fetch the URL url
INFO: Creating context with default config
```

- For non-HTTP(S) URL locations, after you specify the URL, you can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to the URL location. (HTTP(S) is read only).
- The admin context file must be stored on the internal flash memory.
- Available URL types include: **disk***number* (for flash memory), **ftp**, **http**, **https**, or **tftp**.
- To change the URL, reenter the config-url command with a new URL.

Step 7 (Optional) Assign a context to a failover group in Active/Active failover:

join-failover-group {**1** | **2**}

Example:

```
ciscoasa(config-ctx)# join-failover-group 2
```

By default, contexts are in group 1. The admin context must always be in group 1.

Step 8 (Optional) Enable Cloud Web Security for this context:

scansafe [**license** *key*]

Example:

```
ciscoasa(config-ctx)# scansafe
```

If you do not specify a **license**, the context uses the license configured in the system configuration. The ASA sends the authentication key to the Cloud Web Security proxy servers to indicate from which organization the request comes. The authentication key is a 16-byte hexidecimal number.

See the firewall configuration guide for detailed information about ScanSafe.

Example

The following example sets the admin context to be "administrator," creates a context called "administrator" on the internal flash memory, and then adds two contexts from an FTP server:

```
ciscoasa(config)# admin-context admin
ciscoasa(config)# context admin
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.1
ciscoasa(config-ctx)# config-url disk0:/admin.cfg

ciscoasa(config-ctx)# context test
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
ciscoasa(config-ctx)# member gold


ciscoasa(config-ctx)# context sample
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
int3-int8
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
ciscoasa(config-ctx)# member silver
```

Assign MAC Addresses to Context Interfaces Automatically

This section describes how to configure auto-generation of MAC addresses. The MAC address is used to classify packets within a context.

Before you begin

- When you configure a **nameif** command for the interface in a context, the new MAC address is generated immediately. If you enable this feature after you configure context interfaces, then MAC addresses are generated for all interfaces immediately after you enable it. If you disable this feature, the MAC address for each interface reverts to the default MAC address. For example, subinterfaces of GigabitEthernet 0/1 revert to using the MAC address of GigabitEthernet 0/1.
- In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface within the context.

Procedure

Automatically assign private MAC addresses to each context interface:

mac-address auto [**prefix** *prefix*]

Example:

```
ciscoasa(config)# mac-address auto prefix 19
```

If you do not enter a prefix, then the ASA autogenerates the prefix based on the last two bytes of the interface (ASA 5500-X) or backplane (ASASM) MAC address.

If you manually enter a prefix, then the *prefix* is a decimal value between 0 and 65535. This prefix is converted to a four-digit hexadecimal number, and used as part of the MAC address.

Change Between Contexts and the System Execution Space

If you log in to the system execution space (or the admin context), you can change between contexts and perform configuration and monitoring tasks within each context. The running configuration that you edit in a configuration mode, or that is used in the **copy** or **write** commands, depends on your location. When you are in the system execution space, the running configuration consists only of the system configuration; when you are in a context, the running configuration consists only of that context. For example, you cannot view all running configurations (system plus all contexts) by entering the **show running-config** command. Only the current configuration displays.

Procedure

Step 1 Change to a context:

changeto context *name*

The prompt changes to ciscoasa/*name*#

Step 2 Change to the system execution space:

changeto system

The prompt changes to ciscoasa#

Manage Security Contexts

This section describes how to manage security contexts.

Remove a Security Context

You cannot remove the current admin context, unless you remove all contexts using the **clear context** command.

Note If you use failover, there is a delay between when you remove the context on the active unit and when the context is removed on the standby unit. You might see an error message indicating that the number of interfaces on the active and standby units are not consistent; this error is temporary and can be ignored.

Before you begin

Perform this procedure in the system execution space.

Procedure

Step 1 Remove a single context:

no context *name*

All context commands are also removed. The context configuration file is not removed from the config URL location.

Step 2 Remove all contexts (including the admin context):

clear context

The context configuration files are not removed from the config URL locations.

Change the Admin Context

The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users.

Before you begin

- You can set any context to be the admin context, as long as the configuration file is stored in the internal flash memory.
- Perform this procedure in the system execution space.

Procedure

Set the admin context:

admin-context *context_name*

Example:

```
ciscoasa(config)# admin-context administrator
```

Any remote management sessions, such as Telnet, SSH, or HTTPS, that are connected to the admin context are terminated. You must reconnect to the new admin context.

A few system configuration commands, including **ntp server**, identify an interface name that belongs to the admin context. If you change the admin context, and that interface name does not exist in the new admin context, be sure to update any system commands that refer to the interface.

Change the Security Context URL

This section describes how to change the context URL.

Before you begin

- You cannot change the security context URL without reloading the configuration from the new URL. The ASA merges the new configuration with the current running configuration.
- Reentering the same URL also merges the saved configuration with the running configuration.
- A merge adds any new commands from the new configuration to the running configuration.
  - If the configurations are the same, no changes occur.
  - If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results. If the running configuration is blank (for example, if the server was unavailable and the configuration was never downloaded), then the new configuration is used.
- If you do not want to merge the configurations, you can clear the running configuration, which disrupts any communications through the context, and then reload the configuration from the new URL.
- Perform this procedure in the system execution space.

Procedure

Step 1 (Optional, if you do not want to perform a merge) Change to the context and clear configuration:

changeto context *name*

clear configure all

Example:

```
ciscoasa(config)# changeto context ctx1
ciscoasa/ctx1(config)# clear configure all
```

If you want to perform a merge, skip to Step 2.

Step 2 Change to the system execution space:

changeto system

Example:

```
ciscoasa/ctx1(config)# changeto system
ciscoasa(config)#
```

Step 3 Enter the context configuration mode for the context you want to change.

context *name*

Example:

```
ciscoasa(config)# context ctx1
```

Step 4 Enter the new URL. The system immediately loads the context so that it is running.

config-url *new_url*

Example:

```
ciscoasa(config)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/ctx1.cfg
```

Reload a Security Context

You can reload the context in two ways:

- Clear the running configuration and then import the startup configuration.

  This action clears most attributes associated with the context, such as connections and NAT tables.

- Remove the context from the system configuration.

  This action clears additional attributes, such as memory allocation, which might be useful for troubleshooting. However, to add the context back to the system requires you to respecify the URL and interfaces.

Reload by Clearing the Configuration

Procedure

Step 1 Change to the context that you want to reload:

changeto context *name*

Example:

```
ciscoasa(config)# changeto context ctx1
```

```
ciscoasa/ctx1(comfig)#
```

Step 2 Clear the running configuration:

clear configure all

This command clears all connections.

Step 3 Reload the configuration:

copy startup-config running-config

Example:

```
ciscoasa/ctx1(config)# copy startup-config running-config
```

The ASA copies the configuration from the URL specified in the system configuration. You cannot change the URL from within a context.

Reload by Removing and Re-adding the Context

To reload the context by removing and then re-adding it, perform the steps.

Procedure

Step 1 Remove a Security Context, on page 195. **Also delete config URL file from the disk**

Step 2 Configure a Security Context, on page 191

Monitoring Security Contexts

This section describes how to view and monitor context information.

View Context Information

From the system execution space, you can view a list of contexts including the name, allocated interfaces, and configuration file URL.

Procedure

Show all contexts:

show context [*name* | **detail**| **count**]

If you want to show information for a particular context, specify the *name*.

The **detail** option shows additional information. See the following sample outputs below for more information.

The **count** option shows the total number of contexts.

Example

The following is sample output from the **show context** command. The following sample output shows three contexts:

```
ciscoasa# show context

Context Name      Interfaces                    URL
*admin            GigabitEthernet0/1.100        disk0:/admin.cfg
                  GigabitEthernet0/1.101
contexta          GigabitEthernet0/1.200        disk0:/contexta.cfg
                  GigabitEthernet0/1.201
contextb          GigabitEthernet0/1.300        disk0:/contextb.cfg
                  GigabitEthernet0/1.301
Total active Security Contexts: 3
```

The following table shows each field description.

Table 6: show context Fields

Field	Description
Context Name	Lists all context names. The context name with the asterisk (*) is the admin context.
Interfaces	The interfaces assigned to the context.
URL	The URL from which the ASA loads the context configuration.

The following is sample output from the **show context detail** command:

```
ciscoasa# show context detail

Context "admin", has been created, but initial ACL rules not complete
  Config URL: disk0:/admin.cfg
  Real Interfaces: Management0/0
  Mapped Interfaces: Management0/0
  Flags: 0x00000013, ID: 1

Context "ctx", has been created, but initial ACL rules not complete
  Config URL: ctx.cfg
  Real Interfaces: GigabitEthernet0/0.10, GigabitEthernet0/1.20,
     GigabitEthernet0/2.30
  Mapped Interfaces: int1, int2, int3
  Flags: 0x00000011, ID: 2

Context "system", is a system resource
  Config URL: startup-config
  Real Interfaces:
  Mapped Interfaces: Control0/0, GigabitEthernet0/0,
     GigabitEthernet0/0.10, GigabitEthernet0/1, GigabitEthernet0/1.10,
     GigabitEthernet0/1.20, GigabitEthernet0/2, GigabitEthernet0/2.30,
     GigabitEthernet0/3, Management0/0, Management0/0.1
  Flags: 0x00000019, ID: 257

Context "null", is a system resource
  Config URL: ... null ...
  Real Interfaces:
  Mapped Interfaces:
  Flags: 0x00000009, ID: 258
```

See the command reference for more information about the **detail** output.

The following is sample output from the **show context count** command:

```
ciscoasa# show context count
Total active contexts: 2
```

View Resource Allocation

From the system execution space, you can view the allocation for each resource across all classes and class members.

Procedure

Show the resource allocation:

show resource allocation [**detail**]

This command shows the resource allocation, but does not show the actual resources being used. See View Resource Usage, on page 204 for more information about actual resource usage.

The **detail** argument shows additional information. See the following sample outputs for more information.

Example

The following sample output shows the total allocation of each resource as an absolute value and as a percentage of the available system resources:

```
ciscoasa# show resource allocation
Resource                    Total       % of Avail
 Conns [rate]               35000           N/A
 Inspects [rate]            35000           N/A
 Syslogs [rate]             10500           N/A
 Conns                     305000           30.50%
 Hosts                      78842           N/A
 SSH                           35           35.00%
 Routes                      5000           N/A
 Telnet                        35           35.00%
 Xlates                     91749           N/A

 Other VPN Sessions            20           2.66%
 Other VPN Burst               20           2.66%
 All                    unlimited
```

The following table shows each field description.

Table 7: show resource allocation Fields

Field	Description
Resource	The name of the resource that you can limit.
Total	The total amount of the resource that is allocated across all contexts. The amount is an absolute number of concurrent instances or instances per second. If you specified a percentage in the class definition, the ASA converts the percentage to an absolute number for this display.
% of Avail	The percentage of the total system resources that is allocated across all contexts, if the resource has a hard system limit. If a resource does not have a system limit, this column shows N/A.

The following is sample output from the **show resource allocation detail** command:

```
ciscoasa# show resource allocation detail
Resource Origin:
    A    Value was derived from the resource 'all'
    C    Value set in the definition of this class
    D    Value set in default class
Resource         Class          Mmbrs  Origin      Limit      Total    Total %
Conns [rate]     default          all      CA  unlimited
                 gold               1       C      34000      34000     N/A
                 silver             1      CA      17000      17000     N/A
                 bronze             0      CA       8500
                 All Contexts:      3                         51000     N/A

Inspects [rate]  default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      10000      10000     N/A
                 bronze             0      CA       5000
                 All Contexts:      3                         10000     N/A

Syslogs [rate]   default          all      CA  unlimited
                 gold               1       C       6000       6000     N/A
                 silver             1      CA       3000       3000     N/A
                 bronze             0      CA       1500
                 All Contexts:      3                          9000     N/A

Conns            default          all      CA  unlimited
                 gold               1       C     200000     200000     20.00%
                 silver             1      CA     100000     100000     10.00%
                 bronze             0      CA      50000
                 All Contexts:      3                        300000     30.00%

Hosts            default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      26214      26214      N/A
                 bronze             0      CA      13107
                 All Contexts:      3                         26214      N/A
```

```
SSH              default          all       C          5
                 gold               1       D          5          5      5.00%
                 silver             1      CA         10         10     10.00%
                 bronze             0      CA          5
                 All Contexts:      3                            20     20.00%

Telnet           default          all       C          5
                 gold               1       D          5          5      5.00%
                 silver             1      CA         10         10     10.00%
                 bronze             0      CA          5
                 All Contexts:      3                            20     20.00%

Routes           default          all       C  unlimited                 N/A
                 gold               1       D  unlimited          5      N/A
                 silver             1      CA         10         10      N/A
                 bronze             0      CA          5                 N/A
                 All Contexts:      3                            20      N/A

Xlates           default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      23040      23040     N/A
                 bronze             0      CA      11520
                 All Contexts:      3                         23040     N/A

mac-addresses    default          all       C      65535
                 gold               1       D      65535      65535    100.00%
                 silver             1      CA       6553       6553      9.99%
                 bronze             0      CA       3276
                 All Contexts:      3                        137623    209.99%
```

The following table shows each field description.

Table 8: show resource allocation detail Fields

Field	Description
Resource	The name of the resource that you can limit.
Class	The name of each class, including the default class. The All contexts field shows the total values across all classes.
Mmbrs	The number of contexts assigned to each class.
Origin	The origin of the resource limit, as follows: • A—You set this limit with the **all** option, instead of as an individual resource. • C—This limit is derived from the member class. • D—This limit was not defined in the member class, but was derived from the default class. For a context assigned to the default class, the value will be “C” instead of “D.” The ASA can combine “A” with “C” or “D.”

Field	Description
Limit	The limit of the resource per context, as an absolute number. If you specified a percentage in the class definition, the ASA converts the percentage to an absolute number for this display.
Total	The total amount of the resource that is allocated across all contexts in the class. The amount is an absolute number of concurrent instances or instances per second. If the resource is unlimited, this display is blank.
% of Avail	The percentage of the total system resources that is allocated across all contexts in the class. If the resource is unlimited, this display is blank. If the resource does not have a system limit, then this column shows N/A.

View Resource Usage

From the system execution space, you can view the resource usage for each context and display the system resource usage.

Procedure

View resource usage for each context:

show resource usage [**context** *context_name* | **top** *n* | **all** | **summary** | **system**] [**resource** {*resource_name* | **all**} | **detail**] [**counter** *counter_name* [*count_threshold*]]

- By default, **all** context usage is displayed; each context is listed separately.
- Enter the **top** *n* keyword to show the contexts that are the top *n* users of the specified resource. You must specify a single resource type, and not **resource all**, with this option.
- The **summary** option shows all context usage combined.
- The **system** option shows all context usage combined, but shows the system limits for resources instead of the combined context limits.
- For the **resource** *resource_name*, see Configure a Class for Resource Management, on page 187 for available resource names. See also the **show resource type** command. Specify **all** (the default) for all types.
- The **detail** option shows the resource usage of all resources, including those you cannot manage. For example, you can view the number of TCP intercepts.
- The **counter** *counter_name* is one of the following keywords:
    - **current**—Shows the active concurrent instances or the current rate of the resource.

- **denied**—Shows the number of instances that were denied because they exceeded the resource limit shown in the Limit column.
- **peak**—Shows the peak concurrent instances, or the peak rate of the resource since the statistics were last cleared, either using the **clear resource usage** command or because the device rebooted.
- **all**—(Default) Shows all statistics.

- The *count_threshold* sets the number above which resources are shown. The default is 1. If the usage of the resource is below the number you set, then the resource is not shown. If you specify **all** for the counter name, then the *count_threshold* applies to the current usage.
- To show all resources, set the *count_threshold* to **0**.

Examples

The following is sample output from the **show resource usage context** command, which shows the resource usage for the admin context:

```
ciscoasa# show resource usage context admin

Resource              Current         Peak      Limit    Denied  Context
Telnet                      1            1          5         0  admin
Conns                      44           55        N/A         0  admin
Hosts                      45           56        N/A         0  admin
```

The following is sample output from the **show resource usage summary** command, which shows the resource usage for all contexts and all resources. This sample shows the limits for six contexts.

```
ciscoasa# show resource usage summary

Resource              Current         Peak      Limit     Denied Context
Syslogs [rate]           1743         2132      N/A            0 Summary
Conns                     584          763      280000(S)      0 Summary
Xlates                   8526         8966      N/A            0 Summary
Hosts                     254          254      N/A            0 Summary
Conns [rate]              270          535      N/A         1704 Summary
Inspects [rate]           270          535      N/A            0 Summary

Other VPN Sessions        0             10      10           740 Summary
Other VPN Burst           0             10      10           730 Summary

S = System: Combined context limits exceed the system limit; the system limit is shown.
```

The following is sample output from the **show resource usage summary** command, which shows the limits for 25 contexts. Because the context limit for Telnet and SSH connections is 5 per context, then the combined limit is 125. The system limit is only 100, so the system limit is shown.

```
ciscoasa# show resource usage summary

Resource              Current         Peak      Limit    Denied   Context
Telnet                      1            1        100[S]      0   Summary
SSH                         2            2        100[S]      0   Summary
```

```
Conns                     56          90        130000(S)   0   Summary
Hosts                     89         102        N/A         0   Summary
S = System: Combined context limits exceed the system limit; the system limit is shown.
```

The following is sample output from the **show resource usage system** command, which shows the resource usage for all contexts, but it shows the system limit instead of the combined context limits. The **counter all 0** option is used to show resources that are not currently in use. The Denied statistics indicate how many times the resource was denied due to the system limit, if available.

```
ciscoasa# show resource usage system counter all 0

Resource              Current         Peak      Limit        Denied    Context
Telnet                      0            0        100             0    System
SSH                         0            0        100             0    System
ASDM                        0            0         32             0    System
Routes                      0            0        N/A             0    System
IPSec                       0            0          5             0    System
Syslogs [rate]              1           18        N/A             0    System
Conns                       0            1     280000             0    System
Xlates                      0            0        N/A             0    System
Hosts                       0            2        N/A             0    System
Conns [rate]                1            1        N/A             0    System
Inspects [rate]             0            0        N/A             0    System

Other VPN Sessions          0           10        750           740    System
Other VPN Burst             0           10        750           730    System
```

Monitor SYN Attacks in Contexts

The ASA prevents SYN attacks using TCP Intercept. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the ASA acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the ASA receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

Procedure

Step 1 Monitor the rate of attacks for individual contexts:

show perfmon

Step 2 Monitor the amount of resources being used by TCP intercept for individual contexts:

show resource usage detail

Step 3 Monitor the resources being used by TCP intercept for the entire system:

show resource usage summary detail

Examples

The following is sample output from the **show perfmon** command that shows the rate of TCP intercepts for a context called admin.

```
ciscoasa/admin# show perfmon

Context:admin
PERFMON STATS:   Current      Average
Xlates               0/s          0/s
Connections          0/s          0/s
TCP Conns            0/s          0/s
UDP Conns            0/s          0/s
URL Access           0/s          0/s
URL Server Req       0/s          0/s
WebSns Req           0/s          0/s
TCP Fixup            0/s          0/s
HTTP Fixup           0/s          0/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
TCP Intercept    322779/s      322779/s
```

The following is sample output from the **show resource usage detail** command that shows the amount of resources being used by TCP Intercept for individual contexts. (Sample text in **bold** shows the TCP intercept information.)

```
ciscoasa(config)# show resource usage detail
Resource              Current         Peak      Limit        Denied Context
memory                 843732       847288  unlimited             0 admin
chunk:channels             14           15  unlimited             0 admin
chunk:fixup                15           15  unlimited             0 admin
chunk:hole                  1            1  unlimited             0 admin
chunk:ip-users             10           10  unlimited             0 admin
chunk:list-elem            21           21  unlimited             0 admin
chunk:list-hdr              3            4  unlimited             0 admin
chunk:route                 2            2  unlimited             0 admin
chunk:static                1            1  unlimited             0 admin
tcp-intercepts         328787       803610  unlimited             0 admin
np-statics                  3            3  unlimited             0 admin
statics                     1            1  unlimited             0 admin
ace-rules                   1            1  unlimited             0 admin
console-access-rul          2            2  unlimited             0 admin
fixup-rules                14           15  unlimited             0 admin
memory                 959872       960000  unlimited             0 c1
chunk:channels             15           16  unlimited             0 c1
chunk:dbgtrace              1            1  unlimited             0 c1
chunk:fixup                15           15  unlimited             0 c1
chunk:global                1            1  unlimited             0 c1
chunk:hole                  2            2  unlimited             0 c1
chunk:ip-users             10           10  unlimited             0 c1
chunk:udp-ctrl-blk          1            1  unlimited             0 c1
chunk:list-elem            24           24  unlimited             0 c1
chunk:list-hdr              5            6  unlimited             0 c1
chunk:nat                   1            1  unlimited             0 c1
chunk:route                 2            2  unlimited             0 c1
chunk:static                1            1  unlimited             0 c1
tcp-intercept-rate      16056        16254  unlimited             0 c1
globals                     1            1  unlimited             0 c1
```

```
np-statics                  3            3  unlimited             0 c1
statics                     1            1  unlimited             0 c1
nats                        1            1  unlimited             0 c1
ace-rules                   2            2  unlimited             0 c1
console-access-rul          2            2  unlimited             0 c1
fixup-rules                14           15  unlimited             0 c1
memory              232695716    232020648  unlimited             0 system
chunk:channels             17           20  unlimited             0 system
chunk:dbgtrace              3            3  unlimited             0 system
chunk:fixup                15           15  unlimited             0 system
chunk:ip-users              4            4  unlimited             0 system
chunk:list-elem          1014         1014  unlimited             0 system
chunk:list-hdr              1            1  unlimited             0 system
chunk:route                 1            1  unlimited             0 system
block:16384               510          885  unlimited             0 system
block:2048                 32           34  unlimited             0 system
```

The following sample output shows the resources being used by TCP intercept for the entire system. (Sample text in **bold** shows the TCP intercept information.)

```
ciscoasa(config)# show resource usage summary detail
Resource              Current         Peak      Limit        Denied Context
memory              238421312    238434336  unlimited             0 Summary
chunk:channels             46           48  unlimited             0 Summary
chunk:dbgtrace              4            4  unlimited             0 Summary
chunk:fixup                45           45  unlimited             0 Summary
chunk:global                1            1  unlimited             0 Summary
chunk:hole                  3            3  unlimited             0 Summary
chunk:ip-users             24           24  unlimited             0 Summary
chunk:udp-ctrl-blk          1            1  unlimited             0 Summary
chunk:list-elem          1059         1059  unlimited             0 Summary
chunk:list-hdr             10           11  unlimited             0 Summary
chunk:nat                   1            1  unlimited             0 Summary
chunk:route                 5            5  unlimited             0 Summary
chunk:static                2            2  unlimited             0 Summary
block:16384               510          885  unlimited             0 Summary
block:2048                 32           35  unlimited             0 Summary
tcp-intercept-rate     341306       811579  unlimited             0 Summary
globals                     1            1  unlimited             0 Summary
np-statics                  6            6  unlimited             0 Summary
statics                     2            2        N/A             0 Summary
nats                        1            1        N/A             0 Summary
ace-rules                   3            3        N/A             0 Summary
console-access-rul          4            4        N/A             0 Summary
fixup-rules                43           44        N/A             0 Summary
```

View Assigned MAC Addresses

You can view auto-generated MAC addresses within the system configuration or within the context.

View MAC Addresses in the System Configuration

This section describes how to view MAC addresses in the system configuration.

Before you begin

If you manually assign a MAC address to an interface, but also have auto-generation enabled, the auto-generated address continues to show in the configuration even though the manual MAC address is the one that is in use. If you later remove the manual MAC address, the auto-generated one shown will be used.

Procedure

Show the assigned MAC addresses from the system execution space:

show running-config all context [*name*]

The **all** option is required to view the assigned MAC addresses. Although the **mac-address auto** command is user-configurable in global configuration mode only, the command appears as a read-only entry in context configuration mode along with the assigned MAC address. Only allocated interfaces that are configured with a **nameif** command within the context have a MAC address assigned.

Examples

The following output from the **show running-config all context admin** command shows the primary and standby MAC address assigned to the Management0/0 interface:

```
ciscoasa# show running-config all context admin

context admin
  allocate-interface Management0/0
  mac-address auto Management0/0 a24d.0000.1440 a24d.0000.1441
  config-url disk0:/admin.cfg
```

The following output from the **show running-config all context** command shows all the MAC addresses (primary and standby) for all context interfaces. Note that because the GigabitEthernet0/0 and GigabitEthernet0/1 main interfaces are not configured with a **nameif** command inside the contexts, no MAC addresses have been generated for them.

```
ciscoasa# show running-config all context

admin-context admin
context admin
  allocate-interface Management0/0
  mac-address auto Management0/0 a2d2.0400.125a a2d2.0400.125b
  config-url disk0:/admin.cfg
!

context CTX1
  allocate-interface GigabitEthernet0/0
  allocate-interface GigabitEthernet0/0.1-GigabitEthernet0/0.5
  mac-address auto GigabitEthernet0/0.1 a2d2.0400.11bc a2d2.0400.11bd
  mac-address auto GigabitEthernet0/0.2 a2d2.0400.11c0 a2d2.0400.11c1
  mac-address auto GigabitEthernet0/0.3 a2d2.0400.11c4 a2d2.0400.11c5
  mac-address auto GigabitEthernet0/0.4 a2d2.0400.11c8 a2d2.0400.11c9
  mac-address auto GigabitEthernet0/0.5 a2d2.0400.11cc a2d2.0400.11cd
  allocate-interface GigabitEthernet0/1
  allocate-interface GigabitEthernet0/1.1-GigabitEthernet0/1.3
  mac-address auto GigabitEthernet0/1.1 a2d2.0400.120c a2d2.0400.120d
  mac-address auto GigabitEthernet0/1.2 a2d2.0400.1210 a2d2.0400.1211
  mac-address auto GigabitEthernet0/1.3 a2d2.0400.1214 a2d2.0400.1215
  config-url disk0:/CTX1.cfg
!

context CTX2
  allocate-interface GigabitEthernet0/0
  allocate-interface GigabitEthernet0/0.1-GigabitEthernet0/0.5
  mac-address auto GigabitEthernet0/0.1 a2d2.0400.11ba a2d2.0400.11bb
  mac-address auto GigabitEthernet0/0.2 a2d2.0400.11be a2d2.0400.11bf
  mac-address auto GigabitEthernet0/0.3 a2d2.0400.11c2 a2d2.0400.11c3
  mac-address auto GigabitEthernet0/0.4 a2d2.0400.11c6 a2d2.0400.11c7
  mac-address auto GigabitEthernet0/0.5 a2d2.0400.11ca a2d2.0400.11cb
  allocate-interface GigabitEthernet0/1
  allocate-interface GigabitEthernet0/1.1-GigabitEthernet0/1.3
  mac-address auto GigabitEthernet0/1.1 a2d2.0400.120a a2d2.0400.120b
  mac-address auto GigabitEthernet0/1.2 a2d2.0400.120e a2d2.0400.120f
  mac-address auto GigabitEthernet0/1.3 a2d2.0400.1212 a2d2.0400.1213
  config-url disk0:/CTX2.cfg
!
```

View MAC Addresses Within a Context

This section describes how to view MAC addresses within a context.

Procedure

Show the MAC address in use by each interface within the context:

show interface | include (Interface)|(MAC)

Example

For example:

```
ciscoasa/context# show interface | include (Interface)|(MAC)

Interface GigabitEthernet1/1.1 "g1/1.1", is down, line protocol is down
        MAC address a201.0101.0600, MTU 1500
Interface GigabitEthernet1/1.2 "g1/1.2", is down, line protocol is down
        MAC address a201.0102.0600, MTU 1500
Interface GigabitEthernet1/1.3 "g1/1.3", is down, line protocol is down
        MAC address a201.0103.0600, MTU 1500
...
```

Note The **show interface** command shows the MAC address in use; if you manually assign a MAC address and also have auto-generation enabled, then you can only view the unused auto-generated address from within the system configuration.

Examples for Multiple Context Mode

The following example:

- Automatically sets the MAC addresses in contexts with a custom prefix.
- Sets the default class limit for conns to 10 percent instead of unlimited, and sets the VPN other sessions to 10, with a burst of 5.
- Creates a gold resource class.
- Sets the admin context to be “administrator.”
- Creates a context called “administrator” on the internal flash memory to be part of the default resource class.
- Adds two contexts from an FTP server as part of the gold resource class.

```
ciscoasa(config)# mac-address auto prefix 19

ciscoasa(config)# class default
ciscoasa(config-class)# limit-resource conns 10%
ciscoasa(config-class)# limit-resource vpn other 10
ciscoasa(config-class)# limit-resource vpn burst other 5

ciscoasa(config)# class gold
ciscoasa(config-class)# limit-resource mac-addresses 10000
ciscoasa(config-class)# limit-resource conns 15%
ciscoasa(config-class)# limit-resource rate conns 1000
ciscoasa(config-class)# limit-resource rate inspects 500
ciscoasa(config-class)# limit-resource hosts 9000
ciscoasa(config-class)# limit-resource asdm 5
ciscoasa(config-class)# limit-resource ssh 5
ciscoasa(config-class)# limit-resource rate syslogs 5000
ciscoasa(config-class)# limit-resource telnet 5
ciscoasa(config-class)# limit-resource xlates 36000
ciscoasa(config-class)# limit-resource routes 700
ciscoasa(config-class)# limit-resource vpn other 100
ciscoasa(config-class)# limit-resource vpn burst other 50

ciscoasa(config)# admin-context administrator
ciscoasa(config)# context administrator
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.1
ciscoasa(config-ctx)# config-url disk0:/admin.cfg

ciscoasa(config-ctx)# context test
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
ciscoasa(config-ctx)# member gold

ciscoasa(config-ctx)# context sample
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
```

```
int3-int8
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
ciscoasa(config-ctx)# member gold
```

History for Multiple Context Mode

Table 9: History for Multiple Context Mode

Feature Name	Platform Releases	Feature Information
Multiple security contexts	7.0(1)	Multiple context mode was introduced. We introduced the following commands: **context**, **mode**, and **class**.
Automatic MAC address assignment	7.2(1)	Automatic assignment of MAC address to context interfaces was introduced. We introduced the following command: **mac-address auto**.
Resource management	7.2(1)	Resource management was introduced. We introduced the following commands: **class**, **limit-resource**, and **member**.
Virtual sensors for IPS	8.0(2)	The AIP SSM running IPS software Version 6.0 and above can run multiple virtual sensors, which means you can configure multiple security policies on the AIP SSM. You can assign each context or single mode ASA to one or more virtual sensors, or you can assign multiple security contexts to the same virtual sensor. We introduced the following command: **allocate-ips**.
Automatic MAC address assignment enhancements	8.0(5)/8.2(2)	The MAC address format was changed to use a prefix, to use a fixed starting value (A2), and to use a different scheme for the primary and secondary unit MAC addresses in a failover pair. The MAC addresses are also now persistent across reloads. The command parser now checks if auto-generation is enabled; if you want to also manually assign a MAC address, you cannot start the manual MAC address with A2. We modified the following command: **mac-address auto prefix**.
Maximum contexts increased for the ASA 5550 and 5580	8.4(1)	The maximum security contexts for the ASA 5550 was increased from 50 to 100. The maximum for the ASA 5580 was increased from 50 to 250.
Automatic MAC address assignment enabled by default	8.5(1)	Automatic MAC address assignment is now enabled by default. We modified the following command: **mac-address auto**.

Feature Name	Platform Releases	Feature Information
Automatic generation of a MAC address prefix	8.6(1)	In multiple context mode, the ASA now converts the automatic MAC address generation configuration to use a default prefix. The ASA auto-generates the prefix based on the last two bytes of the interface (ASA 5500-X) or backplane (ASASM) MAC address. This conversion happens automatically when you reload, or if you reenable MAC address generation. The prefix method of generation provides many benefits, including a better guarantee of unique MAC addresses on a segment. You can view the auto-generated prefix by entering the **show running-config mac-address** command. If you want to change the prefix, you can reconfigure the feature with a custom prefix. The legacy method of MAC address generation is no longer available. **Note** To maintain hitless upgrade for failover pairs, the ASA does *not* convert the MAC address method in an existing configuration upon a reload if failover is enabled. However, we strongly recommend that you manually change to the prefix method of generation when using failover, especially for the ASASM. Without the prefix method, ASASMs installed in different slot numbers experience a MAC address change upon failover, and can experience traffic interruption. After upgrading, to use the prefix method of MAC address generation, reenable MAC address generation to use the default prefix. We modified the following command: **mac-address auto**.
Automatic MAC address assignment disabled by default on all models except for the ASASM	9.0(1)	Automatic MAC address assignment is now disabled by default except for the ASASM. We modified the following command: **mac-address auto**.
Dynamic routing in Security Contexts	9.0(1)	EIGRP and OSPFv2 dynamic routing protocols are now supported in multiple context mode. OSPFv3, RIP, and multicast routing are not supported.
New resource type for routing table entries	9.0(1)	A new resource type, routes, was created to set the maximum number of routing table entries in each context. We modified the following commands: **limit-resource, show resource types, show resource usage, show resource allocation**.
Site-to-Site VPN in multiple context mode	9.0(1)	Site-to-site VPN tunnels are now supported in multiple context mode.
New resource type for site-to-site VPN tunnels	9.0(1)	New resource types, vpn other and vpn burst other, were created to set the maximum number of site-to-site VPN tunnels in each context. We modified the following commands: **limit-resource, show resource types, show resource usage, show resource allocation**.
New resource type for IKEv1 SA negotiations	9.1(2)	New resource type, ikev1 in-negotiation, was created to set the maximum percentage of IKEv1 SA negotiations in each context to prevent overwhelming the CPU and crypto engines. Under certain conditions (large certificates, CRL checking), you might want to restrict this resource. We modified the following commands: **limit-resource, show resource types, show resource usage, show resource allocation**.

CHAPTER 8

Failover for High Availability

This chapter describes how to configure Active/Standby or Active/Active failover to accomplish high availability of the Cisco ASA.

About Failover

Configuring failover requires two identical ASAs connected to each other through a dedicated failover link and, optionally, a state link. The health of the active units and interfaces is monitored to determine whether they meet the specific failover conditions. If those conditions are met, failover occurs.

Failover Modes

The ASA supports two failover modes, Active/Active failover and Active/Standby failover. Each failover mode has its own method for determining and performing failover.

- In Active/Standby failover, one device functions as the Active unit and passes traffic. The second device, designated as the Standby unit, does not actively pass traffic. When a failover occurs, the Active unit fails over to the Standby unit, which then becomes Active. You can use Active/Standby failover for ASAs in single or multiple context mode.

- In an Active/Active failover configuration, both ASAs can pass network traffic. Active/Active failover is only available to ASAs in multiple context mode. In Active/Active failover, you divide the security contexts on the ASA into 2 *failover groups*. A failover group is simply a logical group of one or more security contexts. One group is assigned to be Active on the primary ASA, and the other group is assigned to be active on the Secondary ASA. When a failover occurs, it occurs at the failover group level.

Both failover modes support stateful or stateless failover.

Failover System Requirements

This section describes the hardware, software, and license requirements for ASAs in a Failover configuration.

Hardware Requirements

The two units in a Failover configuration must:

- Be the same model.
- Have the same number and types of interfaces.

  For the Firepower 9300 chassis, all interfaces must be preconfigured in FXOS identically before you enable Failover. If you change the interfaces after you enable Failover, make the interface changes in FXOS on the Standby unit, and then make the same changes on the Active unit. If you remove an interface in FXOS (for example, if you remove a network module, remove an EtherChannel, or reassign an interface to an EtherChannel), then the ASA configuration retains the original commands so that you can make any necessary adjustments; removing an interface from the configuration can have wide effects. You can manually remove the old interface configuration in the ASA OS.
- Have the same modules installed (if any).
- Have the same RAM installed.

If you are using units with different flash memory sizes in your Failover configuration, make sure the unit with the smaller flash memory has enough space to accommodate the software image files and the configuration files. If it does not, configuration synchronization from the unit with the larger flash memory to the unit with the smaller flash memory will fail.

Software Requirements

The two units in a Failover configuration must:

- Be in the same context mode (single or multiple).
- For single mode: Be in the same firewall mode (routed or transparent).

  In multiple context mode, the firewall mode is set at the context-level, and you can use mixed modes.
- Have the same major (first number) and minor (second number) software version. However, you can temporarily use different versions of the software during an upgrade process; for example, you can upgrade one unit from Version 8.3(1) to Version 8.3(2) and have failover remain active. We recommend upgrading both units to the same version to ensure long-term compatibility.
- Have the same AnyConnect images. If the failover pair has mismatched images when a hitless upgrade is performed, then the clientless SSL VPN connection terminates in the final reboot step of the upgrade process, the database shows an orphaned session, and the IP pool shows that the IP address assigned to the client is “in use.”

License Requirements

The two units in a failover configuration do not need to have identical licenses; the licenses combine to make a failover cluster license.

Failover and Stateful Failover Links

The failover link and the optional stateful failover link are dedicated connections between the two units. Cisco recommends to use the same interface between two devices in a failover link or a stateful failover link. For example, in a failover link, if you have used eth0 in device 1, use the same interface (eth0) in device 2 as well.

Caution All information sent over the failover and state links is sent in clear text unless you secure the communication with an IPsec tunnel or a failover key. If the ASA is used to terminate VPN tunnels, this information includes any usernames, passwords and preshared keys used for establishing the tunnels. Transmitting this sensitive data in clear text could pose a significant security risk. We recommend securing the failover communication with an IPsec tunnel or a failover key if you are using the ASA to terminate VPN tunnels.

Failover Link

The two units in a failover pair constantly communicate over a failover link to determine the operating status of each unit.

Failover Link Data

The following information is communicated over the failover link:

- The unit state (active or standby)
- Hello messages (keep-alives)
- Network link status
- MAC address exchange
- Configuration replication and synchronization

Interface for the Failover Link

You can use an unused data interface (physical, subinterface, redundant, or EtherChannel) as the failover link; however, you cannot specify an interface that is currently configured with a name. The failover link interface is not configured as a normal networking interface; it exists for failover communication only. This interface can only be used for the failover link (and also for the state link). For most models, you cannot use a management interface for failover unless explicitly described below.

The ASA does not support sharing interfaces between user data and the failover link. You also cannot use separate subinterfaces on the same parent for the failover link and for data.

See the following guidelines for the failover link:

- 5506-X through 5555-X—You cannot use the Management interface as the failover link; you must use a data interface.
- 5585-X—Do not use the Management 0/0 interface, even though it can be used as a data interface. It does not support the necessary performance for this use.
- Firepower 4100/9300—We recommend that you use a 10 GB data interface for the combined failover and state link. You cannot use the management-type interface for the failover link.
- All other models—1 GB interface is large enough for a combined failover and state link.

For a redundant interface used as the failover link, see the following benefits for added redundancy:

- When a failover unit boots up, it alternates between the member interfaces to detect an active unit.
- If a failover unit stops receiving keepalive messages from its peer on one of the member interfaces, it switches to the other member interface.

The alternation frequency is equal to the unit hold time (the **failover polltime unit** command).

Note If you have a large configuration and a low unit hold time, alternating between the member interfaces can prevent the secondary unit from joining/re-joining. In this case, disable one of the member interfaces until after the secondary unit joins.

For an EtherChannel used as the failover link, to prevent out-of-order packets, only one interface in the EtherChannel is used. If that interface fails, then the next interface in the EtherChannel is used. You cannot alter the EtherChannel configuration while it is in use as a failover link.

Connecting the Failover Link

Connect the failover link in one of the following two ways:

- Using a switch, with no other device on the same network segment (broadcast domain or VLAN) as the failover interfaces of the ASA.
- Using an Ethernet cable to connect the units directly, without the need for an external switch.

If you do not use a switch between the units, if the interface fails, the link is brought down on both peers. This condition may hamper troubleshooting efforts because you cannot easily determine which unit has the failed interface and caused the link to come down.

The ASA supports Auto-MDI/MDIX on its copper Ethernet ports, so you can either use a crossover cable or a straight-through cable. If you use a straight-through cable, the interface automatically detects the cable and swaps one of the transmit/receive pairs to MDIX.

Stateful Failover Link

To use Stateful Failover, you must configure a Stateful Failover link (also known as the state link) to pass connection state information.

Shared with the Failover Link

Sharing a failover link is the best way to conserve interfaces. However, you must consider a dedicated interface for the state link and failover link, if you have a large configuration and a high traffic network.

Dedicated Interface

You can use a dedicated data interface (physical, redundant, or EtherChannel) for the state link. See Interface for the Failover Link, on page 217 for requirements for a dedicated state link, and Connecting the Failover Link, on page 218 for information about connecting the state link as well.

For optimum performance when using long distance failover, the latency for the state link should be less than 10 milliseconds and no more than 250 milliseconds. If latency is more than 10 milliseconds, some performance degradation occurs due to retransmission of failover messages.

Avoiding Interrupted Failover and Data Links

We recommend that failover links and data interfaces travel through different paths to decrease the chance that all interfaces fail at the same time. If the failover link is down, the ASA can use the data interfaces to determine if a failover is required. Subsequently, the failover operation is suspended until the health of the failover link is restored.

See the following connection scenarios to design a resilient failover network.

Scenario 1—Not Recommended

If a single switch or a set of switches are used to connect both failover and data interfaces between two ASAs, then when a switch or inter-switch-link is down, both ASAs become active. Therefore, the following two connection methods shown in the following figures are NOT recommended.

Figure 33: Connecting with a Single Switch—Not Recommended

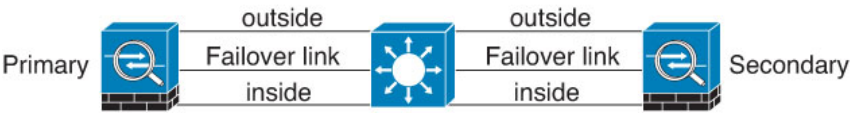
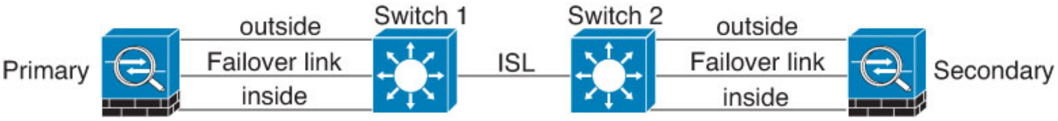


Figure 34: Connecting with a Double-Switch—Not Recommended



Scenario 2—Recommended

We recommend that failover links NOT use the same switch as the data interfaces. Instead, use a different switch or use a direct cable to connect the failover link, as shown in the following figures.

Figure 35: Connecting with a Different Switch

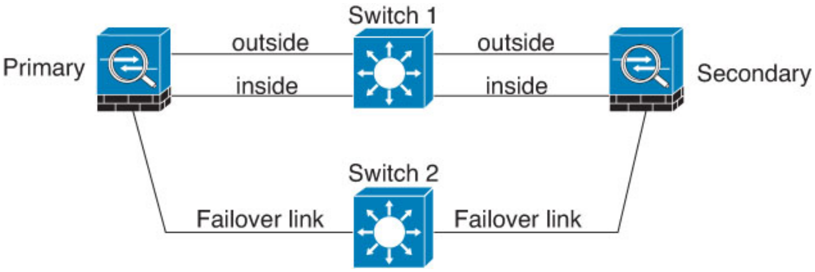
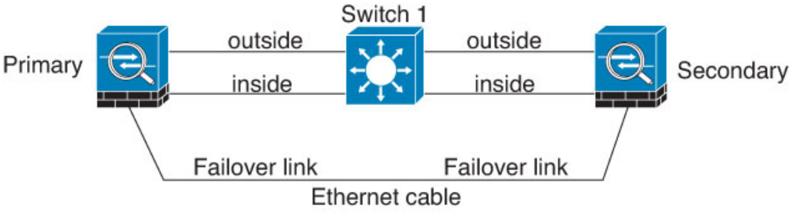


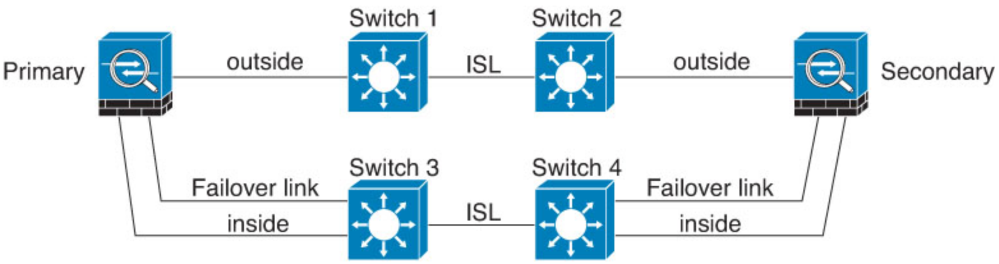
Figure 36: Connecting with a Cable



Scenario 3—Recommended

If the ASA data interfaces are connected to more than one set of switches, then a failover link can be connected to one of the switches, preferably the switch on the secure (inside) side of network, as shown in the following figure.

Figure 37: Connecting with a Secure Switch



Scenario 4—Recommended

The most reliable failover configurations use a redundant interface on the failover link, as shown in the following figures.

Figure 38: Connecting with Redundant Interfaces

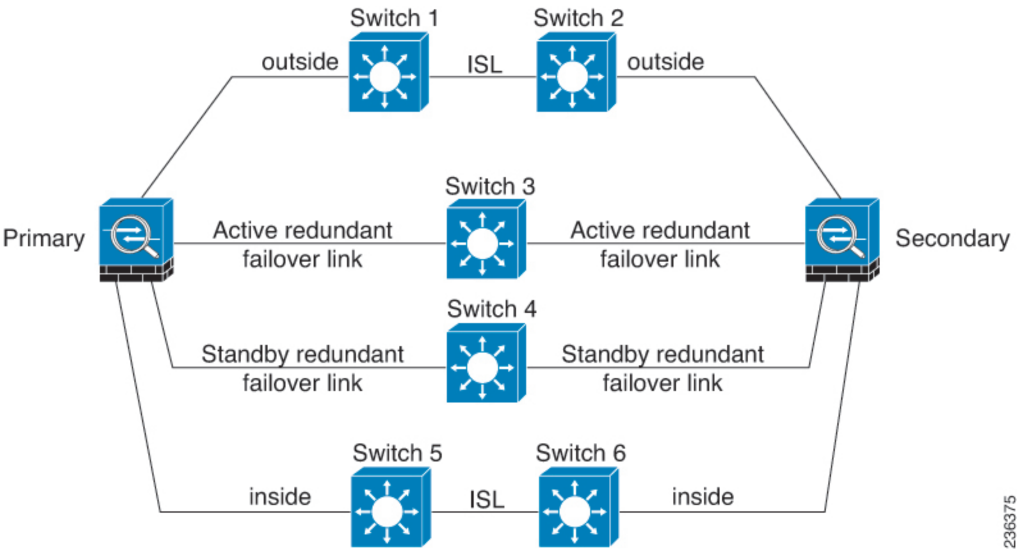
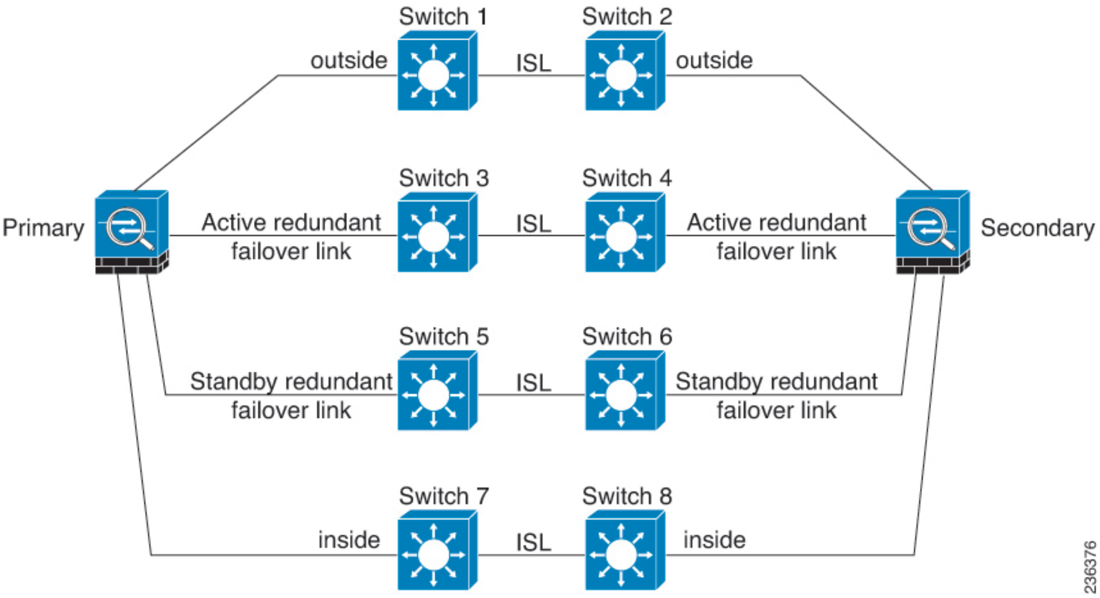

Switch 1
Switch 2
outside
ISL
outside
Switch 3
Primary
Active redundant failover link
Active redundant failover link
Secondary
Switch 4
Standby redundant failover link
Standby redundant failover link
Switch 5
Switch 6
inside
ISL
inside
236375

Figure 39: Connecting with Inter-switch Links



MAC Addresses and IP Addresses in Failover

When you configure your interfaces, you can specify an active IP address and a standby IP address on the same network. Generally, when a failover occurs, the new active unit takes over the active IP addresses and MAC addresses. Because network devices see no change in the MAC to IP address pairing, no ARP entries change or time out anywhere on the network.

Note Although recommended, the standby address is not required. Without a standby IP address, the active unit cannot perform network tests to check the standby interface health; it can only track the link state. You also cannot connect to the standby unit on that interface for management purposes.

The IP address and MAC address for the state link do not change at failover.

Active/Standby IP Addresses and MAC Addresses

For Active/Standby Failover, see the following for IP address and MAC address usage during a failover event:

1. The active unit always uses the primary unit's IP addresses and MAC addresses.
2. When the active unit fails over, the standby unit assumes the IP addresses and MAC addresses of the failed unit and begins passing traffic.
3. When the failed unit comes back online, it is now in a standby state and takes over the standby IP addresses and MAC addresses.

However, if the secondary unit boots without detecting the primary unit, then the secondary unit becomes the active unit and uses its own MAC addresses, because it does not know the primary unit MAC addresses. When the primary unit becomes available, the secondary (active) unit changes the MAC addresses to those of the primary unit, which can cause an interruption in your network traffic. Similarly, if you swap out the primary unit with new hardware, a new MAC address is used.

Virtual MAC addresses guard against this disruption, because the active MAC addresses are known to the secondary unit at startup, and remain the same in the case of new primary unit hardware. If you do not configure virtual MAC addresses, you might need to clear the ARP tables on connected routers to restore traffic flow. The ASA does not send gratuitous ARPs for static NAT addresses when the MAC address changes, so connected routers do not learn of the MAC address change for these addresses.

Active/Active IP Addresses and MAC Addresses

For Active/Active failover, see the following for IP address and MAC address usage during a failover event:

1. The primary unit autogenerates active and standby MAC addresses for all interfaces in failover group 1 and 2 contexts. You can also manually configure the MAC addresses if necessary, for example, if there are MAC address conflicts.
2. Each unit uses the active IP addresses and MAC addresses for its active failover group, and the standby addresses for its standby failover group. For example, the primary unit is active for failover group 1, so it uses the active addresses for contexts in failover group 1. It is standby for the contexts in failover group 2, where it uses the standby addresses.
3. When a unit fails over, the other unit assumes the active IP addresses and MAC addresses of the failed failover group and begins passing traffic.
4. When the failed unit comes back online, and you enabled the preempt option, it resumes the failover group.

Virtual MAC Addresses

The ASA has multiple methods to configure virtual MAC addresses. We recommend using only one method. If you set the MAC address using multiple methods, the MAC address used depends on many variables, and might not be predictable. Manual methods include the interface mode **mac-address** command, the **failover mac address** command, and for Active/Active failover, the failover group mode **mac address** command, in addition to autogeneration methods described below.

In multiple context mode, you can configure the ASA to generate virtual active and standby MAC addresses automatically for shared interfaces, and these assignments are synced to the secondary unit (see the **mac-address auto** command). For non-shared interfaces, you can manually set the MAC addresses for Active/Standby mode (Active/Active mode autogenerates MAC addresses for all interfaces).

For Active/Active failover, virtual MAC addresses are always used, either with default values or with values you can set per interface.

Intra- and Inter-Chassis Module Placement for the ASA Services Module

You can place the primary and secondary ASASMs within the same switch or in two separate switches.

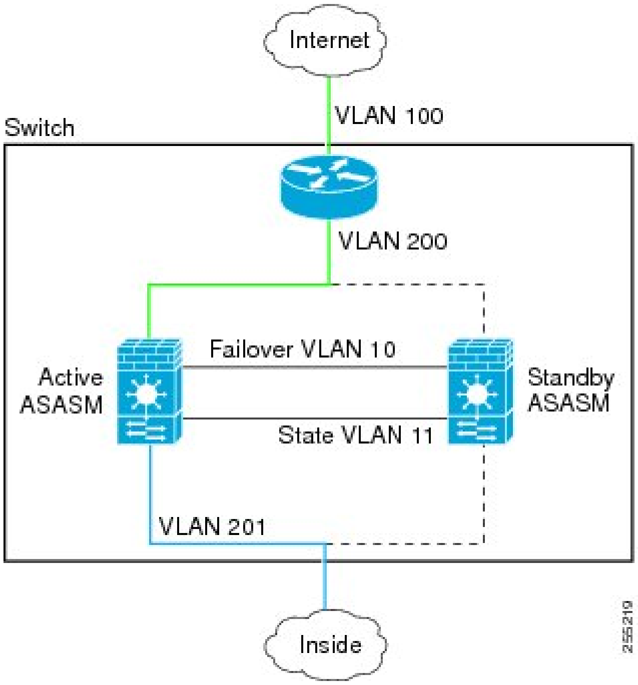
Intra-Chassis Failover

If you install the secondary ASASM in the same switch as the primary ASASM, you protect against module-level failure.

Even though both ASASMs are assigned the same VLANs, only the active module takes part in networking. The standby module does not pass any traffic.

The following figure shows a typical intra-switch configuration.

Figure 40: Intra-Switch Failover



Inter-Chassis Failover

To protect against switch-level failure, you can install the secondary ASASM in a separate switch. The ASASM does not coordinate failover directly with the switch, but it works harmoniously with the switch failover operation. See the switch documentation to configure failover for the switch.

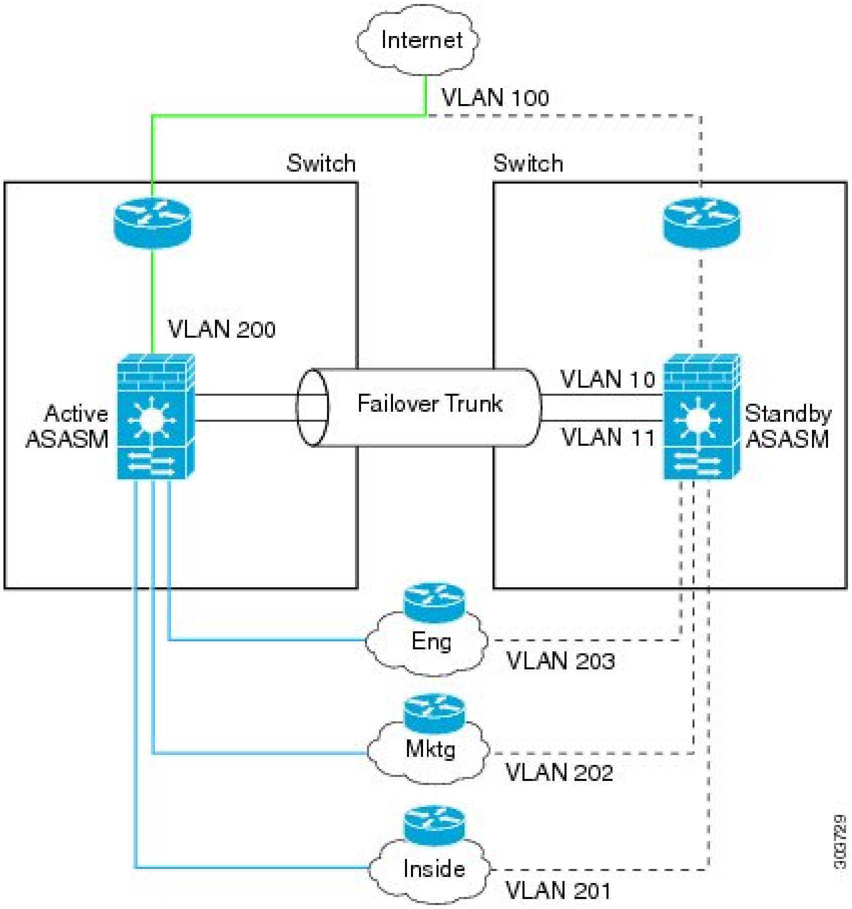
For the best reliability of failover communications between ASASMs, we recommend that you configure an EtherChannel trunk port between the two switches to carry the failover and state VLANs.

For other VLANs, you must ensure that both switches have access to all firewall VLANs, and that monitored VLANs can successfully pass hello packets between both switches.

The following figure shows a typical switch and ASASM redundancy configuration. The trunk between the two switches carries the failover ASASM VLANs (VLANs 10 and 11).

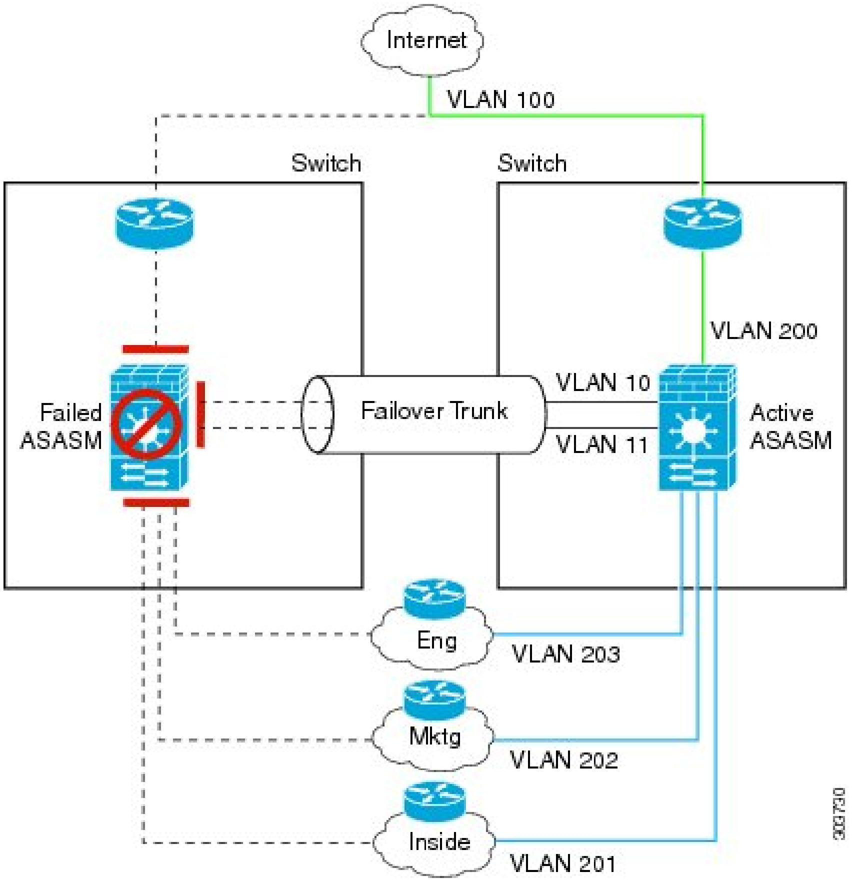
Note ASASM failover is independent of the switch failover operation; however, ASASM works in any switch failover scenario.

Figure 41: Normal Operation



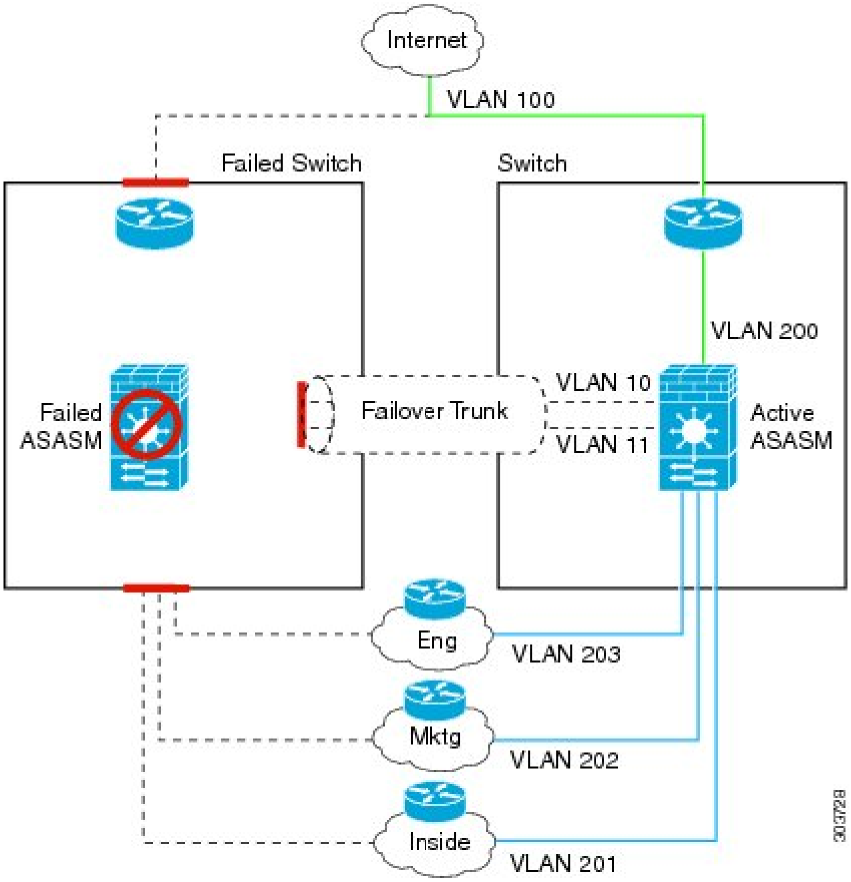
If the primary ASASM fails, then the secondary ASASM becomes active and successfully passes the firewall VLANs.

Figure 42: ASASM Failure



If the entire switch fails, as well as the ASASM (such as in a power failure), then both the switch and the ASASM fail over to their secondary units.

Figure 43: Switch Failure



Stateless and Stateful Failover

The ASA supports two types of failover, stateless and stateful for both the Active/Standby and Active/Active modes.

Note Some configuration elements for clientless SSL VPN (such as bookmarks and customization) use the VPN failover subsystem, which is part of Stateful Failover. You must use Stateful Failover to synchronize these elements between the members of the failover pair. Stateless failover is not recommended for clientless SSL VPN.

Stateless Failover

When a failover occurs, all active connections are dropped. Clients need to reestablish connections when the new active unit takes over.

Note Some configuration elements for clientless SSL VPN (such as bookmarks and customization) use the VPN failover subsystem, which is part of Stateful Failover. You must use Stateful Failover to synchronize these elements between the members of the failover pair. Stateless (regular) failover is not recommended for clientless SSL VPN.

Stateful Failover

When Stateful Failover is enabled, the active unit continually passes per-connection state information to the standby unit, or in Active/Active failover, between the active and standby failover groups. After a failover occurs, the same connection information is available at the new active unit. Supported end-user applications are not required to reconnect to keep the same communication session.

Supported Features

For Stateful Failover, the following state information is passed to the standby ASA:

- NAT translation table.
- TCP and UDP connections and states. Other types of IP protocols, and ICMP, are not parsed by the active unit, because they get established on the new active unit when a new packet arrives.
- The HTTP connection table (unless you enable HTTP replication).
- The HTTP connection states (if HTTP replication is enabled)—By default, the ASA does not replicate HTTP session information when Stateful Failover is enabled. We suggest that you enable HTTP replication.
- The ARP table
- The Layer 2 bridge table (for bridge groups)
- The ISAKMP and IPsec SA table
- GTP PDP connection database
- SIP signaling sessions and pin holes.
- ICMP connection state—ICMP connection replication is enabled only if the respective interface is assigned to an asymmetric routing group.
- Static and dynamic routing tables—Stateful Failover participates in dynamic routing protocols, like OSPF and EIGRP, so routes that are learned through dynamic routing protocols on the active unit are maintained in a Routing Information Base (RIB) table on the standby unit. Upon a failover event, packets travel normally with minimal disruption to traffic because the active secondary unit initially has rules that mirror the primary unit. Immediately after failover, the re-convergence timer starts on the newly active unit. Then the epoch number for the RIB table increments. During re-convergence, OSPF and EIGRP routes become updated with a new epoch number. Once the timer is expired, stale route entries (determined by the epoch number) are removed from the table. The RIB then contains the newest routing protocol forwarding information on the newly active unit.

Note Routes are synchronized only for link-up or link-down events on an active unit. If the link goes up or down on the standby unit, dynamic routes sent from the active unit may be lost. This is normal, expected behavior.

- DHCP Server—DHCP address leases are not replicated. However, a DHCP server configured on an interface will send a ping to make sure an address is not being used before granting the address to a DHCP client, so there is no impact to the service. State information is not relevant for DHCP relay or DDNS.
- Cisco IP SoftPhone sessions—If a failover occurs during an active Cisco IP SoftPhone session, the call remains active because the call session state information is replicated to the standby unit. When the call is terminated, the IP SoftPhone client loses connection with the Cisco Call Manager. This connection loss occurs because there is no session information for the CTIQBE hangup message on the standby unit. When the IP SoftPhone client does not receive a response back from the Call Manager within a certain time period, it considers the Call Manager unreachable and unregisters itself.
- RA VPN—Remote access VPN end users do not have to reauthenticate or reconnect the VPN session after a failover. However, applications operating over the VPN connection could lose packets during the failover process and not recover from the packet loss.

Unsupported Features

For Stateful Failover, the following state information is not passed to the standby ASA:

- The user authentication (uauth) table
- TCP state bypass connections
- Multicast routing.
- State information for modules, such as the ASA FirePOWER module.
- Selected clientless SSL VPN features:
  - Smart Tunnels
  - Port Forwarding
  - Plugins
  - Java Applets
  - IPv6 clientless or Anyconnect sessions
  - Citrix authentication (Citrix users must reauthenticate after failover)

Transparent Firewall Mode Bridge Group Requirements for Failover

There are special considerations for failover when using bridge groups.

Transparent Mode Bridge Group Requirements for Appliances, ASAv

When the active unit fails over to the standby unit, the connected switch port running Spanning Tree Protocol (STP) can go into a blocking state for 30 to 50 seconds when it senses the topology change. To avoid traffic loss while the port is in a blocking state, you can configure one of the following workarounds depending on the switch port mode:

- Access mode—Enable the STP PortFast feature on the switch:

  ```
  interface interface_id
    spanning-tree portfast
  ```

  The PortFast feature immediately transitions the port into STP forwarding mode upon linkup. The port still participates in STP. So if the port is to be a part of the loop, the port eventually transitions into STP blocking mode.

- Trunk mode—Block BPDUs on the ASA on a bridge group's member interfaces with an EtherType access rule.

  ```
  access-list id ethertype deny bpdu
  access-group id in interface name1
  access-group id in interface name2
  ```

  Blocking BPDUs disables STP on the switch. Be sure not to have any loops involving the ASA in your network layout.

If neither of the above options are possible, then you can use one of the following less desirable workarounds that impacts failover functionality or STP stability:

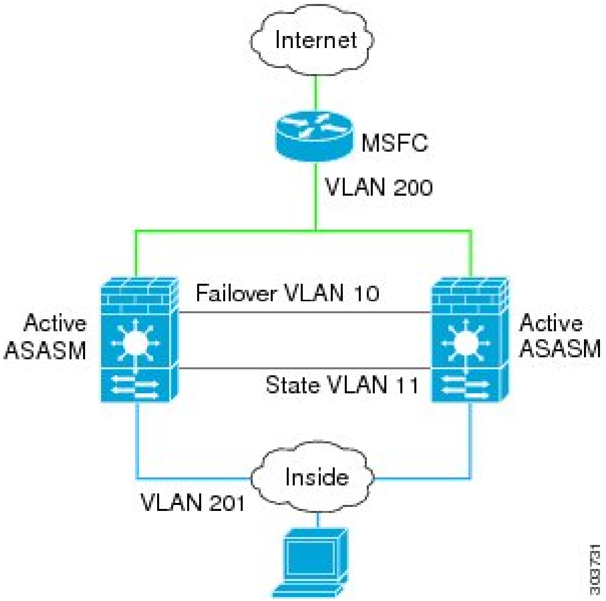
- Disable interface monitoring.
- Increase interface holdtime to a high value that will allow STP to converge before the ASAs fail over.
- Decrease STP timers to allow STP to converge faster than the interface holdtime.

Transparent Mode Bridge Group Requirements for the ASA Services Module

To avoid loops when you use failover with bridge groups, you should allow BPDUs to pass (the default), and you must use switch software that supports BPDU forwarding.

Loops can occur if both modules are active at the same time, such as when both modules are discovering each other’s presence, or due to a bad failover link. Because the ASASMs bridge packets between the same two VLANs, loops can occur when packets between bridge group member interfaces get endlessly replicated by both ASASMs. The spanning tree protocol can break such loops if there is a timely exchange of BPDUs. To break the loop, BPDUs sent between VLAN 200 and VLAN 201 need to be bridged.

Figure 44: Bridge Group Loop



Failover Health Monitoring

The ASA monitors each unit for overall health and for interface health. This section includes information about how the ASA performs tests to determine the state of each unit.

Unit Health Monitoring

The ASA determines the health of the other unit by monitoring the failover link with hello messages. When a unit does not receive three consecutive hello messages on the failover link, the unit sends LANTEST messages on each data interface, including the failover link, to validate whether or not the peer is responsive. The action that the ASA takes depends on the response from the other unit. See the following possible actions:

- If the ASA receives a response on the failover link, then it does not fail over.
- If the ASA does not receive a response on the failover link, but it does receive a response on a data interface, then the unit does not failover. The failover link is marked as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby while the failover link is down.
- If the ASA does not receive a response on any interface, then the standby unit switches to active mode and classifies the other unit as failed.

Interface Monitoring

You can monitor up to 1025 interfaces (in multiple context mode, divided between all contexts). You should monitor important interfaces. For example in multiple context mode, you might configure one context to monitor a shared interface: because the interface is shared, all contexts benefit from the monitoring.

When a unit does not receive hello messages on a monitored interface for 15 seconds (the default), it runs interface tests. (To change the period, see the **failover polltime interface** command, or for Active/Active failover, the **polltime interface** command) If one of the interface tests fails for an interface, but this same interface on the other unit continues to successfully pass traffic, then the interface is considered to be failed, and the ASA stops running tests.

If the threshold you define for the number of failed interfaces is met (see the **failover interface-policy** command, or for Active/Active failover, the **interface-policy** command), and the active unit has more failed interfaces than the standby unit, then a failover occurs. If an interface fails on both units, then both interfaces go into the “Unknown” state and do not count towards the failover limit defined by failover interface policy.

An interface becomes operational again if it receives any traffic. A failed ASA returns to standby mode if the interface failure threshold is no longer met.

If you have an ASA FirePOWER module, then the ASA also monitors the health of the module over the backplane interface. Failure of the module is considered a unit failure and will trigger failover. This setting is configurable.

If an interface has IPv4 and IPv6 addresses configured on it, the ASA uses the IPv4 addresses to perform the health monitoring. If an interface has only IPv6 addresses configured on it, then the ASA uses IPv6 neighbor discovery instead of ARP to perform the health monitoring tests. For the broadcast ping test, the ASA uses the IPv6 all nodes address (FE02::1).

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Interface Tests

The ASA uses the following interface tests. The duration of each test is approximately 1.5 seconds by default, or 1/16 of the failover interface holdtime(see the **failover polltime interface** command, or for Active/Active failover, the **interface-policy** command).

1. Link Up/Down test—A test of the interface status. If the Link Up/Down test indicates that the interface is down, then the ASA considers it failed, and testing stops. If the status is Up, then the ASA performs the Network Activity test.

2. Network Activity test—A received network activity test. At the start of the test, each unit clears its received packet count for its interfaces. As soon as a unit receives any eligible packets during the test, then the interface is considered operational. If both units receive traffic, then testing stops. If one unit receives traffic and the other unit does not, then the interface on the unit that does not receive traffic is considered failed, and testing stops. If neither unit receives traffic, then the ASA starts the ARP test.

3. ARP test—A test for successful ARP replies. Each unit sends a single ARP request for the IP address in the most recent entry in its ARP table. If the unit receives an ARP reply or other network traffic during the test, then the interface is considered operational. If the unit does not receive an ARP reply, then the ASA sends a single ARP request for the IP address in the *next* entry in the ARP table. If the unit receives an ARP reply or other network traffic during the test, then the interface is considered operational. If both units receive traffic, then testing stops. If one unit receives traffic, and the other unit does not, then the interface on the unit that does not receive traffic is considered failed, and testing stops. If neither unit receives traffic, then the ASA starts the Broadcast Ping test.

4. Broadcast Ping test—A test for successful ping replies. Each unit sends a broadcast ping, and then counts all received packets. If the unit receives any packets during the test, then the interface is considered operational. If both units receive traffic, then testing stops. If one unit receives traffic, and the other unit does not, then the interface on the unit that does not receive traffic is considered failed, and testing stops. If neither unit receives traffic, then testing starts over again with the ARP test. If both units continue to

receive no traffic from the ARP and Broadcast Ping tests, then these tests will continue running in perpetuity.

Interface Status

Monitored interfaces can have the following status:

- Unknown—Initial status. This status can also mean the status cannot be determined.
- Normal—The interface is receiving traffic.
- Testing—Hello messages are not heard on the interface for five poll times.
- Link Down—The interface or VLAN is administratively down.
- No Link—The physical link for the interface is down.
- Failed—No traffic is received on the interface, yet traffic is heard on the peer interface.

Failover Times

The following events trigger failover in a Firepower high availability pair:

- More than 50% of the Snort instances on the active unit are down.
- Disk space on the active unit is more than 90% full.
- The **no failover active** command is run on the active unit or the **failover active** command is run on the standby unit.
- The active unit has more failed interfaces than the standby unit.
- Interface failure on the active device exceeds the threshold configured.

  By default, failure of a single interface causes failover. You can change the default value by configuring a threshold for the number of interfaces or a percentage of monitored interfaces that must fail for the failover to occur. If the threshold breaches on the active device, failover occurs. If the threshold breaches on the standby device, the unit moves to **Fail** state.

  To change the default failover criteria, enter the following command in global configuration mode:

Table 10:

Command	Purpose
failover interface-policy num [%] `hostname (config)# failover interface-policy 20%`	Changes the default failover criteria. When specifying a specific number of interfaces, the *num* argument can be from 1 to 250. When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Note: If you manually fail over using the CLI or ASDM, or you reload the ASA, the failover starts immediately and is not subject to the timers listed below.

Table 11: ASA

Failover Condition	Minimum	Default	Maximum
Active unit loses power, hardware goes down, or the software reloads or crashes. When any of these occur, the monitored interfaces or failover link do not receives any hello message.	800 milliseconds	15 seconds	45 seconds
Active unit main board interface link down.	500 milliseconds	5 seconds	15 seconds
Active unit 4GE module interface link down.	2 seconds	5 seconds	15 seconds
Active unit FirePOWER module fails.	2 seconds	2 seconds	2 seconds
Active unit interface up, but connection problem causes interface testing.	5 seconds	25 seconds	75 seconds

Configuration Synchronization

Failover includes various types of configuration synchronization.

Running Configuration Replication

Running configuration replication occurs when any one or both of the devices in the failover pair boot.

In Active/Standby failover, configurations are always synchronized from the active unit to the standby unit.

In Active/Active failover, whichever unit boots second obtains the running configuration from the unit that boots first, regardless of the primary or secondary designation of the booting unit. After both units are up, commands entered in the system execution space are replicated from the unit on which failover group 1 is in the active state.

When the standby/second unit completes its initial startup, it clears its running configuration (except for the **failover** commands needed to communicate with the active unit), and the active unit sends its entire configuration to the standby/second unit. When the replication starts, the ASA console on the active unit displays the message “Beginning configuration replication: Sending to mate,” and when it is complete, the ASA displays the message “End Configuration Replication to mate.” Depending on the size of the configuration, replication can take from a few seconds to several minutes.

On the unit receiving the configuration, the configuration exists only in running memory. You should save the configuration to flash memory according to Save Configuration Changes, on page 43. For example, in Active/Active failover, enter the **write memory all** command in the system execution space on the unit that has failover group 1 in the active state. The command is replicated to the peer unit, which proceeds to write its configuration to flash memory.

Note During replication, commands entered on the unit sending the configuration may not replicate properly to the peer unit, and commands entered on the unit receiving the configuration may be overwritten by the configuration being received. Avoid entering commands on either unit in the failover pair during the configuration replication process.

Note The **crypto ca server** command and related subcommands are not supported with failover; you must remove them using the **no crypto ca server** command.

File Replication

Configuration syncing does not replicate the following files and configuration components, so you must copy these files manually so they match:

- AnyConnect images
- CSD images
- AnyConnect profiles

  The ASA uses a cached file for the AnyConnect client profile stored in cache:/stc/profiles, and not the file stored in the flash file system. To replicate the AnyConnect client profile to the standby unit, perform one of the following:

  - Enter the **write standby** command on the active unit.
  - Reapply the profile on the active unit.
  - Reload the standby unit.

- Local Certificate Authorities (CAs)
- ASA images
- ASDM images

Command Replication

After startup, commands that you enter on the active unit are immediately replicated on the standby unit. You do not have to save the active configuration to flash memory to replicate the commands.

In Active/Active failover, commands entered in the system execution space are replicated from the unit on which failover group 1 is in the active state.

Failure to enter the commands on the appropriate unit for command replication to occur causes the configurations to be out of synchronization. Those changes may be lost the next time the initial configuration synchronization occurs.

The following commands are replicated to the standby ASA:

- All configuration commands except for **mode**, **firewall**, and **failover lan unit**
- **copy running-config startup-config**
- **delete**
- **mkdir**
- **rename**
- **rmdir**
- **write memory**

The following commands are *not* replicated to the standby ASA:

- All forms of the **copy** command except for **copy running-config startup-config**
- All forms of the **write** command except for **write memory**
- **debug**
- **failover lan unit**
- **firewall**
- **show**
- **terminal pager** and **pager**

About Active/Standby Failover

Active/Standby failover lets you use a standby ASA to take over the functionality of a failed unit. When the active unit fails, the standby unit becomes the active unit.

Note For multiple context mode, the ASA can fail over the entire unit (including all contexts) but cannot fail over individual contexts separately.

Primary/Secondary Roles and Active/Standby Status

The main differences between the two units in a failover pair are related to which unit is active and which unit is standby, namely which IP addresses to use and which unit actively passes traffic.

However, a few differences exist between the units based on which unit is primary (as specified in the configuration) and which unit is secondary:

- The primary unit always becomes the active unit if both units start up at the same time (and are of equal operational health).

- The primary unit MAC addresses are always coupled with the active IP addresses. The exception to this rule occurs when the secondary unit becomes active and cannot obtain the primary unit MAC addresses over the failover link. In this case, the secondary unit MAC addresses are used.

Active Unit Determination at Startup

The active unit is determined by the following:

- If a unit boots and detects a peer already running as active, it becomes the standby unit.
- If a unit boots and does not detect a peer, it becomes the active unit.
- If both units boot simultaneously, then the primary unit becomes the active unit, and the secondary unit becomes the standby unit.

Failover Events

In Active/Standby failover, failover occurs on a unit basis. Even on systems running in multiple context mode, you cannot fail over individual or groups of contexts.

The following table shows the failover action for each failure event. For each failure event, the table shows the failover policy (failover or no failover), the action taken by the active unit, the action taken by the standby unit, and any special notes about the failover condition and actions.

Table 12: Failover Events

Failure Event	Policy	Active Unit Action	Standby Unit Action	Notes
Active unit failed (power or hardware)	Failover	n/a	Become active Mark active as failed	No hello messages are received on any monitored interface or the failover link.
Formerly active unit recovers	No failover	Become standby	No action	None.
Standby unit failed (power or hardware)	No failover	Mark standby as failed	n/a	When the standby unit is marked as failed, then the active unit does not attempt to fail over, even if the interface failure threshold is surpassed.
Failover link failed during operation	No failover	Mark failover link as failed	Mark failover link as failed	You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.
Failover link failed at startup	No failover	Become active Mark failover link as failed	Become active Mark failover link as failed	If the failover link is down at startup, both units become active.

Failure Event	Policy	Active Unit Action	Standby Unit Action	Notes
State link failed	No failover	No action	No action	State information becomes out of date, and sessions are terminated if a failover occurs.
Interface failure on active unit above threshold	Failover	Mark active as failed	Become active	None.
Interface failure on standby unit above threshold	No failover	No action	Mark standby as failed	When the standby unit is marked as failed, then the active unit does not attempt to fail over even if the interface failure threshold is surpassed.

About Active/Active Failover

This section describes Active/Active failover.

Active/Active Failover Overview

In an Active/Active failover configuration, both ASAs can pass network traffic. Active/Active failover is only available to ASAs in multiple context mode. In Active/Active failover, you divide the security contexts on the ASA into a maximum of 2 failover groups.

A failover group is simply a logical group of one or more security contexts. You can assign failover group to be active on the primary ASA, and failover group 2 to be active on the secondary ASA. When a failover occurs, it occurs at the failover group level. For example, depending on interface failure patterns, it is possible for failover group 1 to fail over to the secondary ASA, and subsequently failover group 2 to fail over to the primary ASA. This event could occur if the interfaces in failover group 1 are down on the primary ASA but up on the secondary ASA, while the interfaces in failover group 2 are down on the secondary ASA but up on the primary ASA.

The admin context is always a member of failover group 1. Any unassigned security contexts are also members of failover group 1 by default. If you want Active/Active failover, but are otherwise uninterested in multiple contexts, the simplest configuration would be to add one additional context and assign it to failover group 2.

Note When configuring Active/Active failover, make sure that the combined traffic for both units is within the capacity of each unit.

Note You can assign both failover groups to one ASA if desired, but then you are not taking advantage of having two active ASAs.

Primary/Secondary Roles and Active/Standby Status for a Failover Group

As in Active/Standby failover, one unit in an Active/Active failover pair is designated the primary unit, and the other unit the secondary unit. Unlike Active/Standby failover, this designation does not indicate which unit becomes active when both units start simultaneously. Instead, the primary/secondary designation does two things:

- The primary unit provides the running configuration to the pair when they boot simultaneously.
- Each failover group in the configuration is configured with a primary or secondary unit preference. When used with preemption, this preference ensures that the failover group runs on the correct unit after it starts up. Without preemption, both groups run on the first unit to boot up.

Active Unit Determination for Failover Groups at Startup

The unit on which a failover group becomes active is determined as follows:

- When a unit boots while the peer unit is not available, both failover groups become active on the unit.
- When a unit boots while the peer unit is active (with both failover groups in the active state), the failover groups remain in the active state on the active unit regardless of the primary or secondary preference of the failover group until one of the following occurs:
    - A failover occurs.
    - A failover is manually forced.
    - A preemption for the failover group is configured, which causes the failover group to automatically become active on the preferred unit when the unit becomes available.

Failover Events

In an Active/Active failover configuration, failover occurs on a failover group basis, not a system basis. For example, if you designate both failover groups as Active on the primary unit, and failover group 1 fails, then failover group 2 remains Active on the primary unit while failover group 1 becomes active on the secondary unit.

Because a failover group can contain multiple contexts, and each context can contain multiple interfaces, it is possible for all interfaces in a single context to fail without causing the associated failover group to fail.

The following table shows the failover action for each failure event. For each failure event, the policy (whether or not failover occurs), actions for the active failover group, and actions for the standby failover group are given.

Table 13: Failover Events

Failure Event	Policy	Active Group Action	Standby Group Action	Notes
A unit experiences a power or software failure	Failover	Become standby Mark as failed	Become active Mark active as failed	When a unit in a failover pair fails, any active failover groups on that unit are marked as failed and become active on the peer unit.

Failure Event	Policy	Active Group Action	Standby Group Action	Notes
Interface failure on active failover group above threshold	Failover	Mark active group as failed	Become active	None.
Interface failure on standby failover group above threshold	No failover	No action	Mark standby group as failed	When the standby failover group is marked as failed, the active failover group does not attempt to fail over, even if the interface failure threshold is surpassed.
Formerly active failover group recovers	No failover	No action	No action	Unless failover group preemption is configured, the failover groups remain active on their current unit.
Failover link failed at startup	No failover	Become active	Become active	If the failover link is down at startup, both failover groups on both units become active.
State link failed	No failover	No action	No action	State information becomes out of date, and sessions are terminated if a failover occurs.
Failover link failed during operation	No failover	n/a	n/a	Each unit marks the failover link as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.

Licensing for Failover

Failover units do not require the same license on each unit. If you have licenses on both units, they combine into a single running failover cluster license. There are some exceptions to this rule. See the following table for precise licensing requirements for failover.

Model	License Requirement
ASA 5506-X and ASA 5506W-X	• Active/Standby—Security Plus License. • Active/Active—No Support. **Note** Each unit must have the same encryption license.
ASA 5512-X through ASA 5555-X	• ASA 5512-X—Security Plus License. • Other models—Base License. **Note** • Each unit must have the same encryption license. • In multiple context mode, each unit must have the the same AnyConnect Apex license. • Each unit must have the same IPS module license. You also need the IPS signature subscription on the IPS side for both units. See the following guidelines: • To buy the IPS signature subscription you need to have the ASA with IPS pre-installed (the part number must include "IPS", for example ASA5525-IPS-K9); you cannot buy the IPS signature subscription for a non-IPS part number ASA. • You need the IPS signature subscription on both units; this subscription is not shared in failover, because it is not an ASA license. • The IPS signature subscription requires a unique IPS module license per unit. Like other ASA licenses, the IPS module license is technically shared in the failover cluster license. However, because of the IPS signature subscription requirements, you must buy a separate IPS module license for each unit in.
ASAv	See Failover Licenses for the ASAv, on page 113.
Firepower 9300	See Failover Licenses for the ASA on the Firepower 9300 Chassis, on page 113.
All other models	Base License or Standard License. **Note** • Each unit must have the same encryption license. • In multiple context mode, each unit must have the the same AnyConnect Apex license.

Note A valid permanent key is required; in rare instances, your PAK authentication key can be removed. If your key consists of all 0's, then you need to reinstall a valid authentication key before failover can be enabled.

Guidelines for Failover

Context Mode

- Active/Active mode is supported only in multiple context mode.
- For multiple context mode, perform all steps in the system execution space unless otherwise noted.
- Stateful failover is not supported for AnyConnect connections in multiple context mode.

Model Support

- ASA 5506W-X—You must disable interface monitoring for the internal GigabitEthernet 1/9 interface. These interfaces will not be able to communicate to perform the default interface monitoring checks, resulting in a switch from active to standby and back again because of expected interface communication failures.
- Firepower 9300—We recommend that you use inter-chassis Failover for the best redundancy.
- The ASAv on public cloud networks such as Microsoft Azure and Amazon Web Services are not supported with Failover because Layer 2 connectivity is required.
- The ASA FirePOWER module does not support failover directly; when the ASA fails over, any existing ASA FirePOWER flows are transferred to the new ASA. The ASA FirePOWER module in the new ASA begins inspecting the traffic from that point forward; old inspection states are not transferred.

  You are responsible for maintaining consistent policies on the ASA FirePOWER modules in the high-availability ASA pair to ensure consistent failover behavior.

  Note Create the failover pair before you configure the ASA FirePOWER modules. If the modules are already configured on both devices, clear the interface configuration on the standby device before creating the failover pair. From the CLI on the standby device, enter the **clear configure interface** command.

ASAv Failover for High Availability

When creating a failover pair with the ASAv, it is necessary to add the data interfaces to each ASAv in the same order. If the exact same interfaces are added to each ASAv, but in different order, errors may be presented at the ASAv Console. Failover functionality may also be affected

Additional Guidelines

- When the active unit fails over to the standby unit, the connected switch port running Spanning Tree Protocol (STP) can go into a blocking state for 30 to 50 seconds when it senses the topology change. To avoid traffic loss while the port is in a blocking state, you can enable the STP PortFast feature on the switch:

  interface *interface_id* **spanning-tree portfast**

  This workaround applies to switches connected to both routed mode and bridge group interfaces. The PortFast feature immediately transitions the port into STP forwarding mode upon linkup. The port still participates in STP. So if the port is to be a part of the loop, the port eventually transitions into STP blocking mode.

- You cannot enable failover if a local CA server is configured. Remove the CA configuration using the **no crypto ca server** command.

- Configuring port security on the switches connected to the ASA failover pair can cause communication problems when a failover event occurs. This problem occurs when a secure MAC address configured or learned on one secure port moves to another secure port, a violation is flagged by the switch port security feature.

- You can monitor up to 1025 interfaces on a unit, across all contexts.

- For Active/Standby Failover and a VPN IPsec tunnel, you cannot monitor both the active and standby units using SNMP over the VPN tunnel. The standby unit does not have an active VPN tunnel, and will drop traffic destined for the NMS. You can instead use SNMPv3 with encryption so the IPsec tunnel is not required.

- For Active/Active failover, no two interfaces in the same context should be configured in the same ASR group.

- For Active/Active failover, you can define a maximum of two failover groups.

- For Active/Active failover, when removing failover groups, you must remove failover group 1 last. Failover group1 always contains the admin context. Any context not assigned to a failover group defaults to failover group 1. You cannot remove a failover group that has contexts explicitly assigned to it.

- Immediately after failover, the source address of syslog messages will be the failover interface address for a few seconds.

- When using SNMPv3 with failover, if you replace a failover unit, then SNMPv3 users are not replicated to the new unit. You must re-add the SNMPv3 users to the active unit to force the users to replicate to the new unit; or you can add the users directly on the new unit. Reconfigure each user by entering the **snmp-server user** *username group-name* **v3** command on the active unit or directly to the standby unit with the *priv-password* option and *auth-password* option in their unencrypted forms.

Defaults for Failover

By default, the failover policy consists of the following:

- No HTTP replication in Stateful Failover.

- A single interface failure causes failover.

- The interface poll time is 5 seconds.
- The interface hold time is 25 seconds.
- The unit poll time is 1 second.
- The unit hold time is 15 seconds.
- Virtual MAC addresses are disabled in multiple context mode, except for the ASASM, where they are enabled by default.
- Monitoring on all physical interfaces, or for the ASASM, all VLAN interfaces.

Configure Active/Standby Failover

To configure Active/Standby failover, configure basic failover settings on both the primary and secondary units. All other configuration occurs only on the primary unit, and is then synched to the secondary unit.

Configure the Primary Unit for Active/Standby Failover

Follow the steps in this section to configure the primary in an Active/Standby failover configuration. These steps provide the minimum configuration needed to enable failover on the primary unit.

Before you begin

- We recommend that you configure standby IP addresses for all interfaces except for the failover and state links.
- Do not configure a **nameif** for the failover and state links.
- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Designate this unit as the primary unit:

failover lan unit primary

Step 2 Specify the interface to be used as the failover link:

failover lan interface *if_name interface_id*

Example:

```
ciscoasa(config)# failover lan interface folink gigabitethernet0/3
```

This interface cannot be used for any other purpose (except, optionally, the state link).

The *if_name* argument assigns a name to the interface.

The *interface_id* argument can be a data physical interface, subinterface, redundant interface, or EtherChannel interface ID. On the ASASM, the *interface_id* is a VLAN ID. For the Firepower 9300, you can use any data-type interface.

Step 3 Assign the active and standby IP addresses to the failover link:

failover interface ip *failover_if_name* {*ip_address mask* | *ipv6_address* / *prefix*} **standby** *ip_address*

Example:

```
ciscoasa(config)# failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2
```

Or:

```
ciscoasa(config)# failover interface ip folink 2001:a0a:b00::a0a:b70/64 standby
2001:a0a:b00::a0a:b71
```

This address should be on an unused subnet. 169.254.0.0/16 and fd00:0:0:*::/64 are internally used subnets, and you cannot use them for the failover or state links.

The standby IP address must be in the same subnet as the active IP address.

Step 4 Enable the failover link:

interface *failover_interface_id*

no shutdown

Example:

```
ciscoasa(config)# interface gigabitethernet 0/3
ciscoasa(config-if)# no shutdown
```

Step 5 (Optional) Specify the interface you want to use as the state link:

failover link *if_name interface_id*

Example:

```
ciscoasa(config)# failover link folink gigabitethernet0/3
```

You can share the failover link with the statelink.

The *if_name* argument assigns a name to the interface.

The *interface_id* argument can be a physical interface, subinterface, redundant interface, or EtherChannel interface ID. On the ASASM, the *interface_id* is a VLAN ID.

Step 6 If you specified a separate state link, assign the active and standby IP addresses to the state link:

failover interface ip *state_if_name* {*ip_address mask* | *ipv6_address*/*prefix*} **standby** *ip_address*

Example:

```
ciscoasa(config)# failover interface ip statelink 172.27.49.1 255.255.255.0 standby
172.27.49.2
```

Or:

```
ciscoasa(config)# failover interface ip statelink 2001:a0a:b00:a::a0a:b70/64 standby
2001:a0a:b00:a::a0a:b71
```

This address should be on an unused subnet, different from the failover link. 169.254.0.0/16 and fd00:0:0:*::/64 are internally used subnets, and you cannot use them for the failover or state links.

The standby IP address must be in the same subnet as the active IP address.

Skip this step if you are sharing the state link.

Step 7 If you specified a separate state link, enable the state link.

interface *state_interface_id*

no shutdown

Example:

```
ciscoasa(config)# interface gigabitethernet 0/4
ciscoasa(config-if)# no shutdown
```

Skip this step if you are sharing the state link.

Step 8 (Optional) Do one of the following to encrypt communications on the failover and state links:

- (Preferred) Establish IPsec LAN-to-LAN tunnels on the failover and state links between the units to encrypt all failover communications:

  failover ipsec pre-shared-key [**0** | **8**] *key*

  Example:

  ```
  ciscoasa(config)# failover ipsec pre-shared-key a3rynsun
  ```

  The *key* can be up to 128 characters in length. Identify the same key on both units. The key is used by IKEv2 to establish the tunnels.

  If you use a master passphrase (see Configure the Master Passphrase, on page 509), then the key is encrypted in the configuration. If you are copying from the configuration (for example, from **more system:running-config** output), specify that the key is encrypted by using the **8** keyword. **0** is used by default, specifying an unencrypted password.

  The **failover ipsec pre-shared-key** shows as ***** in **show running-config** output; this obscured key is not copyable.

  If you do not configure failover and state link encryption, failover communication, including any passwords or keys in the configuration that are sent during command replication, will be in clear text.

  You cannot use both IPsec encryption and the legacy **failover key** encryption. If you configure both methods, IPsec is used. However, if you use the master passphrase, you must first remove the failover key using the **no failover key** command before you configure IPsec encryption.

  Failover LAN-to-LAN tunnels do not count against the IPsec (Other VPN) license.

- (Optional) Encrypt failover communication on the failover and state links:

  failover key [**0** | **8**] {**hex** *key* | *shared_secret*}

Example:

```
ciscoasa(config)# failover key johncr1cht0n
```

Use a *shared_secret* from 1 to 63 characters or a 32-character **hex** *key*. For the *shared_secret*, you can use any combination of numbers, letters, or punctuation. The shared secret or hex key is used to generate the encryption key. Identify the same key on both units.

If you use a master passphrase (see Configure the Master Passphrase, on page 509), then the shared secret or hex key is encrypted in the configuration. If you are copying from the configuration (for example, from **more system:running-config** output), specify that the shared secret or hex key is encrypted by using the **8** keyword. **0** is used by default, specifying an unencrypted password.

The **failover key** shared secret shows as ***** in **show running-config** output; this obscured key is not copyable.

If you do not configure failover and state link encryption, failover communication, including any passwords or keys in the configuration that are sent during command replication, will be in clear text.

Step 9 Enable failover:

failover

Step 10 Save the system configuration to flash memory:

write memory

Examples

The following example configures the failover parameters for the primary unit:

```
failover lan unit primary
failover lan interface folink gigabitethernet0/3
failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2

interface gigabitethernet 0/3
  no shutdown
failover link folink gigabitethernet0/3
failover ipsec pre-shared-key a3rynsun
failover
```

Configure the Secondary Unit for Active/Standby Failover

The only configuration required on the secondary unit is for the failover link. The secondary unit requires these commands to communicate initially with the primary unit. After the primary unit sends its configuration to the secondary unit, the only permanent difference between the two configurations is the **failover lan unit** command, which identifies each unit as primary or secondary.

Before you begin

- Do not configure a **nameif** for the failover and state links.

- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Re-enter the exact same commands as on the primary unit *except* for the **failover lan unit primary** command. You can optionally replace it with the **failover lan unit secondary** command, but it is not necessary because **secondary** is the default setting. See Configure the Primary Unit for Active/Standby Failover, on page 243.

For example:

```
ciscoasa(config)# failover lan interface folink gigabitethernet0/3
INFO: Non-failover interface config is cleared on GigabitEthernet0/3 and its sub-interfaces
ciscoasa(config)# failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2
ciscoasa(config)# interface gigabitethernet 0/3
ciscoasa(config-ifc)# no shutdown
ciscoasa(config-ifc)# failover link folink gigabitethernet0/3
ciscoasa(config)# failover ipsec pre-shared-key a3rynsun
ciscoasa(config)# failover
```

Step 2 After the failover configuration syncs, save the configuration to flash memory:

```
ciscoasa(config)# write memory
```

Configure Active/Active Failover

This section tells how to configure Active/Active failover.

Configure the Primary Unit for Active/Active Failover

Follow the steps in this section to configure the primary unit in an Active/Active failover configuration. These steps provide the minimum configuration needed to enable failover on the primary unit.

Before you begin

- Enable multiple context mode according to Enable or Disable Multiple Context Mode, on page 185.
- We recommend that you configure standby IP addresses for all interfaces except for the failover and state links according to Routed and Transparent Mode Interfaces, on page 451.
- Do not configure a **nameif** for the failover and state links.
- Complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Designate this unit as the primary unit:

failover lan unit primary

Step 2 Specify the interface to be used as the failover link:

failover lan interface *if_name interface_id*

Example:

```
ciscoasa(config)# failover lan interface folink gigabitethernet0/3
```

This interface cannot be used for any other purpose (except, optionally, the state link).

The *if_name* argument assigns a name to the interface.

The *interface_id* argument can be a physical interface, subinterface, redundant interface, or EtherChannel interface ID. For the Firepower 9300, you can use any data-type interface.

Step 3 Assign the active and standby IP addresses to the failover link:

standby failover interface ip *if_name* {*ip_address mask* | *ipv6_address*/*prefix* } **standby** *ip_address*

Example:

```
ciscoasa(config)# failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2
```

Or:

```
ciscoasa(config)# failover interface ip folink 2001:a0a:b00::a0a:b70/64 standby
2001:a0a:b00::a0a:b71
```

This address should be on an unused subnet. 169.254.0.0/16 and fd00:0:0:*::/64 are internally used subnets, and you cannot use them for the failover or state links.

The standby IP address must be in the same subnet as the active IP address.

Step 4 Enable the failover link:

interface *failover_interface_id*

no shutdown

Example:

```
ciscoasa(config)# interface gigabitethernet 0/3
ciscoasa(config-if)# no shutdown
```

Step 5 (Optional) Specify the interface you want to use as the state link:

failover link *if_name interface_id*

Example:

```
ciscoasa(config)# failover link statelink gigabitethernet0/4
```

We recommend specifying a separate interface from the failover link or data interfaces.

The *if_name* argument assigns a name to the interface.

The *interface_id* argument can be a physical interface, subinterface, redundant interface, or EtherChannel interface ID. On the ASASM, the *interface_id* specifies a VLAN ID.

Step 6 If you specified a separate state link, assign the active and standby IP addresses to the state link:

This address should be on an unused subnet, different from the failover link. 169.254.0.0/16 and fd00:0:0:*::/64 are internally used subnets, and you cannot use them for the failover or state links.

The standby IP address must be in the same subnet as the active IP address.

Skip this step if you are sharing the state link.

failover interface ip state *if_name* {*ip_address mask* | *ipv6_address*/*prefix*} **standby** *ip_address*

Example:

```
ciscoasa(config)# failover interface ip statelink 172.27.49.1 255.255.255.0 standby
172.27.49.2
```

Or:

```
ciscoasa(config)# failover interface ip statelink 2001:a0a:b00:a::a0a:b70/64 standby
2001:a0a:b00:a::a0a:b71
```

Step 7 If you specified a separate state link, enable the state link:

interface *state_interface_id*

no shutdown

Example:

```
ciscoasa(config)# interface gigabitethernet 0/4
ciscoasa(config-if)# no shutdown
```

Skip this step if you are sharing the state link.

Step 8 (Optional) Do one of the following to encrypt communications on the failover and state links:

- (Preferred) Establish IPsec LAN-to-LAN tunnels on the failover and state links between the units to encrypt all failover communications:

  failover ipsec pre-shared-key [**0** | **8**] *key*

  ```
  ciscoasa(config)# failover ipsec pre-shared-key a3rynsun
  ```

  The *key* can be up to 128 characters in length. Identify the same key on both units. The key is used by IKEv2 to establish the tunnels.

If you use a master passphrase (see Configure the Master Passphrase, on page 509), then the key is encrypted in the configuration. If you are copying from the configuration (for example, from **more system:running-config** output), specify that the key is encrypted by using the **8** keyword. **0** is used by default, specifying an unencrypted password.

The **failover ipsec pre-shared-key** shows as ***** in **show running-config** output; this obscured key is not copyable.

If you do not configure failover and state link encryption, failover communication, including any passwords or keys in the configuration that are sent during command replication, will be in clear text.

You cannot use both IPsec encryption and the legacy **failover key** encryption. If you configure both methods, IPsec is used. However, if you use the master passphrase, you must first remove the failover key using the **no failover key** command before you configure IPsec encryption.

Failover LAN-to-LAN tunnels do not count against the IPsec (Other VPN) license.

- (Optional) Encrypt failover communication on the failover and state links:

  failover key [**0** | **8**] {**hex** *key* | *shared_secret*}

  ```
  ciscoasa(config)# failover key johncr1cht0n
  ```

  Use a *shared_secret*, from 1 to 63 characters, or a 32-character **hex** *key*.

  For the *shared_secret*, you can use any combination of numbers, letters, or punctuation. The shared secret or hex key is used to generate the encryption key. Identify the same key on both units.

  If you use a master passphrase (see Configure the Master Passphrase, on page 509), then the shared secret or hex key is encrypted in the configuration. If you are copying from the configuration (for example, from **more system:running-config** output), specify that the shared secret or hex key is encrypted by using the **8** keyword. **0** is used by default, specifying an unencrypted password.

  The **failover key** shared secret shows as ***** in **show running-config** output; this obscured key is not copyable.

  If you do not configure failover and state link encryption, failover communication, including any passwords or keys in the configuration that are sent during command replication, will be in clear text.

Step 9 Create failover group 1:

failover group 1

primary

preempt [*delay*]

Example:

```
ciscoasa(config-fover-group)# failover group 1
ciscoasa(config-fover-group)# primary
ciscoasa(config-fover-group)# preempt 1200
```

Typically, you assign group 1 to the primary unit, and group 2 to the secondary unit. Both failover groups become active on the unit that boots first (even if it seems like they boot simultaneously, one unit becomes active first), despite the primary or secondary setting for the group. The **preempt** command causes the failover group to become active on the designated unit automatically when that unit becomes available.

You can enter an optional *delay* value, which specifies the number of seconds the failover group remains active on the current unit before automatically becoming active on the designated unit. Valid values are from 1 to 1200.

If Stateful Failover is enabled, the preemption is delayed until the connections are replicated from the unit on which the failover group is currently active.

If you manually fail over, the **preempt** command is ignored.

Step 10 Create failover group 2 and assign it to the secondary unit:

failover group 2

secondary

preempt [*delay*]

Example:

```
ciscoasa(config-fover-group)# failover group 2
ciscoasa(config-fover-group)# secondary
ciscoasa(config-fover-group)# preempt 1200
```

Step 11 Enter the context configuration mode for a given context, and assign the context to a failover group:

context *name*

join-failover-group {**1** | **2**}

Example:

```
ciscoasa(config)# context Eng
ciscoasa(config-ctx)# join-failover-group 2
```

Repeat this command for each context.

Any unassigned contexts are automatically assigned to failover group 1. The admin context is always a member of failover group 1; you cannot assign it to group 2.

Step 12 Enable failover:

failover

Step 13 Save the system configuration to flash memory:

write memory

Examples

The following example configures the failover parameters for the primary unit:

```
failover lan unit primary
failover lan interface folink gigabitethernet0/3
failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2

interface gigabitethernet 0/3
```

```
  no shutdown
failover link statelink gigabitethernet0/4
failover interface ip statelink 172.27.49.1 255.255.255.0 standby 172.27.49.2

interface gigabitethernet 0/4
  no shutdown
failover group 1
  primary
  preempt
failover group 2
  secondary
  preempt
context admin
  join-failover-group 1
failover ipsec pre-shared-key a3rynsun
failover
```

Configure the Secondary Unit for Active/Active Failover

The only configuration required on the secondary unit is for the failover link. The secondary unit requires these commands to communicate initially with the primary unit. After the primary unit sends its configuration to the secondary unit, the only permanent difference between the two configurations is the **failover lan unit** command, which identifies each unit as primary or secondary.

Before you begin

- Enable multiple context mode according to Enable or Disable Multiple Context Mode, on page 185.
- Do not configure a **nameif** for the failover and state links.
- Complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Re-enter the exact same commands as on the primary unit *except* for the **failover lan unit primary** command. You can optionally replace it with the **failover lan unit secondary** command, but it is not necessary because **secondary** is the default setting. You also do not need to enter the **failover group** and **join-failover-group** commands, as they are replicated from the primary unit. See Configure the Primary Unit for Active/Active Failover, on page 247.

For example:

```
ciscoasa(config)# failover lan interface folink gigabitethernet0/3
INFO: Non-failover interface config is cleared on GigabitEthernet0/3 and its sub-interfaces
ciscoasa(config)# failover interface ip folink 172.27.48.1 255.255.255.0 standby 172.27.48.2
ciscoasa(config)# interface gigabitethernet 0/3
no shutdown
ciscoasa(config)# failover link statelink gigabitethernet0/4
INFO: Non-failover interface config is cleared on GigabitEthernet0/4 and its sub-interfaces
ciscoasa(config)# failover interface ip statelink 172.27.49.1 255.255.255.0 standby
172.27.49.2
ciscoasa(config)# interface gigabitethernet 0/4
no shutdown
```

```
ciscoasa(config)# failover ipsec pre-shared-key a3rynsun
ciscoasa(config)# failover
```

Step 2 After the failover configuration syncs from the primary unit, save the configuration to flash memory:

ciscoasa(config)# **write memory**

Step 3 If necessary, force failover group 2 to be active on the secondary unit:

failover active group 2

Configure Optional Failover Parameters

You can customize failover settings as desired.

Configure Failover Criteria and Other Settings

See Defaults for Failover, on page 242 for the default settings for many parameters that you can change in this section. For Active/Active mode, you set most criteria per failover group.

Before you begin

- Configure these settings in the system execution space in multiple context mode.

Procedure

Step 1 Change the unit poll and hold times:

failover polltime [**unit**] [**msec**] *poll_time* [**holdtime** [**msec**] *time*]

Example:

```
ciscoasa(config)# failover polltime unit msec 200 holdtime msec 800
```

The **polltime** range is between 1 and 15 seconds or between 200 and 999 milliseconds. The **holdtime** range is between 1and 45 seconds or between 800 and 999 milliseconds.You cannot enter a holdtime value that is less than 3 times the unit poll time. With a faster poll time, the ASA can detect failure and trigger failover faster. However, faster detection can cause unnecessary switchovers when the network is temporarily congested.

If a unit does not hear hello packet on the failover communication interface for one polling period, additional testing occurs through the remaining interfaces. If there is still no response from the peer unit during the hold time, the unit is considered failed and, if the failed unit is the active unit, the standby unit takes over as the active unit.

In Active/Active mode, you set this rate for the system; you cannot set this rate per failover group.

Step 2 Set the session replication rate in connections per second:

failover replication rate *conns*

Example:

```
ciscoasa(config)# failover replication rate 20000
```

The minimum and maximum rate is determined by your model. The default is the maximum rate. In Active/Active mode, you set this rate for the system; you cannot set this rate per failover group.

Step 3 Disable the ability to make any configuration changes directly on the standby unit or context:

failover standby config-lock

By default, configurations on the standby unit/context are allowed with a warning message.

Step 4 (Active/Active mode only) Specify the failover group you want to customize:

failover group {**1** | **2**}

Example:

```
ciscoasa(config)# failover group 1
ciscoasa(config-fover-group)#
```

Step 5 Enable HTTP state replication:

- For Active/Standby mode:

  failover replication http

- For Active/Active mode:

  replication http

To allow HTTP connections to be included in the state information replication, you need to enable HTTP replication. We recommend enabling HTTP state replication.

Note Because of a delay when deleting HTTP flows from the standby unit when using failover, the **show conn count** output might show different numbers on the active unit vs. the standby unit; if you wait several seconds and re-issue the command, you will see the same count on both units.

Step 6 Set the threshold for failover when interfaces fail:

- For Active/Standby mode:

  failover interface-policy *num* [%]

  Example:

  ```
  ciscoasa (config)# failover interface-policy 20%
  ```

- For Active/Active mode:

  interface-policy *num* [%]

  Example:

  ```
  ciscoasa(config-fover-group)# interface-policy 20%
  ```

By default, one interface failure causes failover.

When specifying a specific number of interfaces, the *num* argument can be from 1 to 1025.

When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Step 7 Change the interface poll and hold times:

- For Active/Standby mode:

  failover polltime interface [**msec**] *polltime* [**holdtime** *time*]

  Example:

  ```
  ciscoasa(config)# failover polltime interface msec 500 holdtime 5
  ```

- For Active/Active mode:

  polltime interface [**msec**] *polltime* [**holdtime***time*]

  Example:

  ```
  ciscoasa(config-fover-group)# polltime interface msec 500 holdtime 5
  ```

- *polltime*—Sets how long to wait between sending a hello packet to the peer. Valid values for the polltime are from 1 to 15 seconds or, if the optional **msec** keyword is used, from 500 to 999 milliseconds. The default is 5 seconds.

- **holdtime***time*—Sets the time (as a calculation) between the last-received hello message from the peer unit and the commencement of interface tests to determine the health of the interface. It also sets the duration of each interface test as holdtime/16. Valid values are from 5 to 75 seconds. The default is 5 times the polltime. You cannot enter a holdtime value that is less than five times the polltime.

  To calculate the time before starting interface tests (y):

  a. x = (*holdtime*/*polltime*)/2, rounded to the nearest integer. (.4 and down rounds down; .5 and up rounds up.)

  b. y = x**polltime*

  For example, if you use the default holdtime of 25 and polltime of 5, then y = 15 seconds.

Step 8 Configure the virtual MAC address for an interface:

- For Active/Standby mode:

  failover mac address *phy_if active_mac standby_mac*

  Example:

  ```
  ciscoasa(config)# failover mac address gigabitethernet0/2 00a0.c969.87c8 00a0.c918.95d8
  ```

- For Active/Active mode:

  mac address *phy_if active_mac standby_mac*

Example:

```
ciscoasa(config-fover-group)# mac address gigabitethernet0/2 00a0.c969.87c8 00a0.c918.95d8
```

The *phy_if* argument is the physical name of the interface, such as gigabitethernet0/1.

The *active_mac* and *standby_mac* arguments are MAC addresses in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE would be entered as 000C.F142.4CDE.

The *active_mac* address is associated with the active IP address for the interface, and the *standby_mac* is associated with the standby IP address for the interface.

You can also set the MAC address using other commands or methods, but we recommend using only one method. If you set the MAC address using multiple methods, the MAC address used depends on many variables, and might not be predictable.

Use the **show interface** command to display the MAC address used by an interface.

Step 9 (Active/Active mode only) Repeat this procedure for the other failover group.

Configure Interface Monitoring

By default, monitoring is enabled on all physical interfaces, or for the ASASM, all VLAN interfaces, and on any hardware or software modules installed on the ASA, such as the ASA FirePOWER module.

You might want to exclude interfaces attached to less critical networks from affecting your failover policy.

You can monitor up to 1025 interfaces on a unit (across all contexts in multiple context mode).

Before you begin

In multiple context mode, configure interfaces within each context.

Procedure

Enable or disable health monitoring for an interface:

[**no**] **monitor-interface** {*if_name* | **service-module**}

Example:

```
ciscoasa(config)# monitor-interface inside
ciscoasa(config)# no monitor-interface eng1
```

If you do not want a hardware or software module failure, such as the ASA FirePOWER module, to trigger failover, you can disable module monitoring using the **no monitor-interface service-module** command. Note that for the ASA 5585-X, if you disable monitoring of the service module, you may also want to disable monitoring of the interfaces on the module, which are monitored separately.

Configure Support for Asymmetrically Routed Packets (Active/Active Mode)

When running in Active/Active failover, a unit might receive a return packet for a connection that originated through its peer unit. Because the ASA that receives the packet does not have any connection information for the packet, the packet is dropped. This drop most commonly occurs when the two ASAs in an Active/Active failover pair are connected to different service providers and the outbound connection does not use a NAT address.

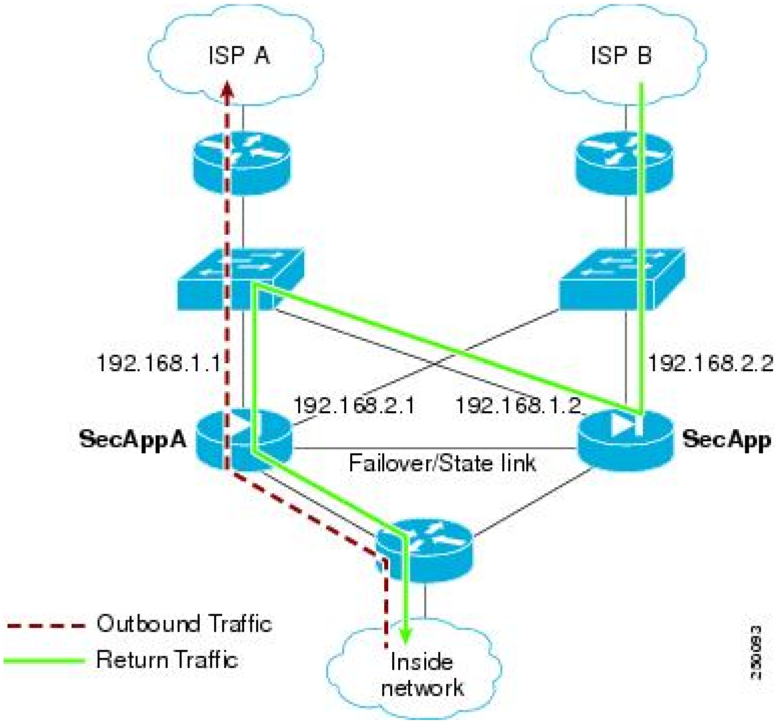
You can prevent the return packets from being dropped by allowing asymmetrically routed packets. To do so, you assign the similar interfaces on each ASA to the same ASR group. For example, both ASAs connect to the inside network on the inside interface, but connect to separate ISPs on the outside interface. On the primary unit, assign the active context outside interface to ASR group 1; on the secondary unit, assign the active context outside interface to the same ASR group 1. When the primary unit outside interface receives a packet for which it has no session information, it checks the session information for the other interfaces in standby contexts that are in the same group; in this case, ASR group 1. If it does not find a match, the packet is dropped. If it finds a match, then one of the following actions occurs:

- If the incoming traffic originated on a peer unit, some or all of the layer 2 header is rewritten and the packet is redirected to the other unit. This redirection continues as long as the session is active.
- If the incoming traffic originated on a different interface on the same unit, some or all of the layer 2 header is rewritten and the packet is reinjected into the stream.

Note This feature does not provide asymmetric routing; it restores asymmetrically routed packets to the correct interface.

The following figure shows an example of an asymmetrically routed packet.

Figure 45: ASR Example



1. An outbound session passes through the ASA with the active SecAppA context. It exits interface outside ISP-A (192.168.1.1).
2. Because of asymmetric routing configured somewhere upstream, the return traffic comes back through the interface outsideISP-B (192.168.2.2) on the ASA with the active SecAppB context.
3. Normally the return traffic would be dropped because there is no session information for the traffic on interface 192.168.2.2. However, the interface is configured as part of ASR group 1. The unit looks for the session on any other interface configured with the same ASR group ID.
4. The session information is found on interface outsideISP-A (192.168.1.2), which is in the standby state on the unit with SecAppB. Stateful Failover replicated the session information from SecAppA to SecAppB.
5. Instead of being dropped, the layer 2 header is rewritten with information for interface 192.168.1.1 and the traffic is redirected out of the interface 192.168.1.2, where it can then return through the interface on the unit from which it originated (192.168.1.1 on SecAppA). This forwarding continues as needed until the session ends.

Before you begin

- Stateful Failover—Passes state information for sessions on interfaces in the active failover group to the standby failover group.
- Replication HTTP—HTTP session state information is not passed to the standby failover group, and therefore is not present on the standby interface. For the ASA to be able to re-route asymmetrically routed HTTP packets, you need to replicate the HTTP state information.
- Perform this procedure within each active context on the primary and secondary units.

- You cannot configure both ASR groups and traffic zones within a context. If you configure a zone in a context, none of the context interfaces can be part of an ASR group.

Procedure

Step 1 On the primary unit, specify the interface for which you want to allow asymmetrically routed packets:

interface *phy_if*

Example:

```
primary/admin(config)# interface gigabitethernet 0/0
```

Step 2 Set the ASR group number for the interface:

asr-group *num*

Example:

```
primary/admin(config-ifc)# asr-group 1
```

Valid values for *num* range from 1 to 32.

Step 3 On the secondary unit, specify the similar interface for which you want to allow asymmetrically routed packets:

interface *phy_if*

Example:

```
secondary/ctx1(config)# interface gigabitethernet 0/1
```

Step 4 Set the ASR group number for the interface to match the primary unit interface:

asr-group *num*

Example:

```
secondary/ctx1(config-ifc)# asr-group 1
```

Examples

The two units have the following configuration (configurations show only the relevant commands). The device labeled SecAppA in the diagram is the primary unit in the failover pair.

Primary Unit System Configuration

```
interface GigabitEthernet0/1
  description LAN/STATE Failover Interface
interface GigabitEthernet0/2
  no shutdown
interface GigabitEthernet0/3
```

```
  no shutdown
interface GigabitEthernet0/4
  no shutdown
interface GigabitEthernet0/5
  no shutdown
failover
failover lan unit primary
failover lan interface folink GigabitEthernet0/1
failover link folink
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
failover group 1
  primary
failover group 2
  secondary
admin-context SecAppA
context admin
  allocate-interface GigabitEthernet0/2
  allocate-interface GigabitEthernet0/3
  config-url flash:/admin.cfg
  join-failover-group 1
context SecAppB
  allocate-interface GigabitEthernet0/4
  allocate-interface GigabitEthernet0/5
  config-url flash:/ctx1.cfg
  join-failover-group 2
```

SecAppA Context Configuration

```
interface GigabitEthernet0/2
  nameif outsideISP-A
  security-level 0
  ip address 192.168.1.1 255.255.255.0 standby 192.168.1.2
  asr-group 1
interface GigabitEthernet0/3
  nameif inside
  security-level 100
  ip address 10.1.0.1 255.255.255.0 standby 10.1.0.11
monitor-interface outside
```

SecAppB Context Configuration

```
interface GigabitEthernet0/4
  nameif outsideISP-B
  security-level 0
  ip address 192.168.2.2 255.255.255.0 standby 192.168.2.1
  asr-group 1
interface GigabitEthernet0/5
  nameif inside
  security-level 100
  ip address 10.2.20.1 255.255.255.0 standby 10.2.20.11
```

Manage Failover

This section describes how to manage Failover units after you enable Failover, including how to change the Failover setup and how to force failover from one unit to another.

Force Failover

To force the standby unit to become active, perform the following procedure.

Before you begin

In multiple context mode, perform this procedure in the System execution space.

Procedure

Step 1 Force a failover when entered on the *standby* unit. The standby unit becomes the active unit.

If you specify the **group** *group_id*, then this command forces a failover when entered on the *standby* unit for the specified Active/Active failover group. The standby unit becomes the active unit for the failover group.

- For Active/Standby mode on the standby unit:

  failover active

- For Active/Active mode on the standby unit:

  failover active [**group** *group_id*]

  Example:

  ```
  standby# failover active group 1
  ```

Step 2 Force a failover when entered on the *active* unit. The active unit becomes the standby unit.

If you specify the **group** *group_id*, then this command forces a failover when entered on the *active* unit for the specified failover group. The active unit becomes the standby unit for the failover group.

- For Active/Standby mode on the active unit:

  no failover active

- For Active/Active mode on the active unit:

  no failover active [**group** *group_id*]

  Example:

  ```
  active# no failover active group 1
  ```

Disable Failover

Disabling failover on one or both units causes the active and standby state of each unit to be maintained until you reload. For an Active/Active failover pair, the failover groups remain in the active state on whichever unit they are active, no matter which unit they are configured to prefer.

See the following characteristics when you disable failover:

- The standby unit/context remains in standby mode so that both units do not start passing traffic (this is called a pseudo-standby state).
- The standby unit/context continues to use its standby IP addresses even though it is no longer connected to an active unit/context.
- The standby unit/context continues to listen for a connection on the failover link. If failover is re-enabled on the active unit/context, then the standby unit/context resumes ordinary standby status after re-synchronizing the rest of its configuration.
- Do not enable failover manually on the standby unit to make it active; instead see Force Failover, on page 261. If you enable failover on the standby unit, you will see a MAC address conflict that can disrupt IPv6 traffic.
- To truly disable failover, save the no failover configuration to the startup configuration, and then reload.

Before you begin

In multiple context mode, perform this procedure in the system execution space.

Procedure

Step 1 Disable failover:

no failover

Step 2 To completely disable failover, save the configuration and reload:

write memory

reload

Restore a Failed Unit

To restore a failed unit to an unfailed state, perform the following procedure.

Before you begin

In multiple context mode, perform this procedure in the System execution space.

Procedure

Step 1 Restore a failed unit to an unfailed state:

- For Active/Standby mode:

  failover reset

- For Active/Active mode:

  failover reset [**group** *group_id*]

Example:

```
ciscoasa(config)# failover reset group 1
```

Restoring a failed unit to an unfailed state does not automatically make it active; restored units remain in the standby state until made active by failover (forced or natural). An exception is a failover group (Active/Active mode only) configured with failover preemption. If previously active, a failover group becomes active if it is configured with preemption and if the unit on which it failed is the preferred unit.

If you specify the **group** *group_id*, this command restores a failed Active/Active failover group to an unfailed state.

Step 2 (Active/Active mode only) To reset failover at the failover group level:

a) In the System choose **Monitoring** > **Failover** > **Failover Group** *#*, where # is the number of the failover group you want to control.

b) Click **Reset Failover**.

Re-Sync the Configuration

If you enter the write standby command on the active unit, the standby unit clears its running configuration (except for the failover commands used to communicate with the active unit), and the active unit sends its entire configuration to the standby unit.

For multiple context mode, when you enter the **write standby** command in the system execution space, all contexts are replicated. If you enter the **write standby** command within a context, the command replicates only the context configuration.

Replicated commands are stored in the running configuration.

Test the Failover Functionality

To test failover functionality, perform the following procedure.

Procedure

Step 1 Test that your active unit is passing traffic as expected by using FTP (for example) to send a file between hosts on different interfaces.

Step 2 Force a failover by entering the following command on the active unit:

Active/Standby mode:

ciscoasa(config)# **no failover active**

Active/Active mode:

ciscoasa(config)# **no failover active group** *group_id*

Step 3 Use FTP to send another file between the same two hosts.

Step 4 If the test was not successful, enter the **show failover** command to check the failover status.

Step 5 When you are finished, you can restore the unit to active status by enter the following command on the newly active unit:

Active/Standby mode:

ciscoasa(config)# **no failover active**

Active/Active mode:

ciscoasa(config)# **failover active group** *group_id*

Note When an ASA interface goes down, for failover it is still considered to be a unit issue. If the ASA detects that an interface is down, failover occurs immediately, without waiting for the interface holdtime. The interface holdtime is only useful when the ASA considers its status to be OK, although it is not receiving hello packets from the peer. To simulate interface holdtime, shut down the VLAN on the switch to prevent peers from receiving hello packets from each other.

Remote Command Execution

Remote command execution lets you send commands entered at the command line to a specific failover peer.

Send a Command

Because configuration commands are replicated from the active unit or context to the standby unit or context, you can use the **failover exec** command to enter configuration commands on the correct unit, no matter which unit you are logged in to. For example, if you are logged in to the standby unit, you can use the **failover exec active** command to send configuration changes to the active unit. Those changes are then replicated to the standby unit. Do not use the **failover exec** command to send configuration commands to the standby unit or context; those configuration changes are not replicated to the active unit and the two configurations will no longer be synchronized.

Output from configuration, exec, and **show** commands is displayed in the current terminal session, so you can use the **failover exec** command to issue **show** commands on a peer unit and view the results in the current terminal.

You must have sufficient privileges to execute a command on the local unit to execute the command on the peer unit.

Procedure

Step 1 If you are in multiple context mode, use the **changeto context***name* command to change to the context you want to configure. You cannot change contexts on the failover peer with the **failover exec** command.

Step 2 Use the following command to send commands to he specified failover unit:

ciscoasa(config)# **failover exec** {**active** | **mate** | **standby**}

Use the **active** or **standby** keyword to cause the command to be executed on the specified unit, even if that unit is the current unit. Use the **mate** keyword to cause the command to be executed on the failover peer.

Commands that cause a command mode change do not change the prompt for the current session. You must use the **show failover exec** command to display the command mode the command is executed in. See Change Command Modes for more information.

Change Command Modes

The **failover exec** command maintains a command mode state that is separate from the command mode of your terminal session. By default, the **failover exec** command mode starts in global configuration mode for the specified device. You can change that command mode by sending the appropriate command (such as the **interface** command) using the **failover exec** command. The session prompt does not change when you change modes using **failover exec**.

For example, if you are logged into global configuration mode of the active unit of a failover pair, and you use the **failover exec active** command to change to interface configuration mode, the terminal prompt remains in global configuration mode, but commands entered using **failover exec** are entered in interface configuration mode.

The following examples show the difference between the terminal session mode and the **failover exec** command mode. In the example, the administrator changes the **failover exec** mode on the active unit to interface configuration mode for the interface GigabitEthernet0/1. After that, all commands entered using **failover exec active** are sent to interface configuration mode for interface GigabitEthernet0/1. The administrator then uses failover exec active to assign an IP address to that interface. Although the prompt indicates global configuration mode, the **failover exec active** mode is in interface configuration mode.

```
ciscoasa(config)# failover exec active interface GigabitEthernet0/1
ciscoasa(config)# failover exec active ip address 192.168.1.1 255.255.255.0 standby
192.168.1.2
ciscoasa(config)# router rip
ciscoasa(config-router)#
```

Changing commands modes for your current session to the device does not affect the command mode used by the **failover exec** command. For example, if you are in interface configuration mode on the active unit, and you have not changed the **failover exec** command mode, the following command would be executed in global configuration mode. The result would be that your session to the device remains in interface configuration mode, while commands entered using **failover exec active** are sent to router configuration mode for the specified routing process.

```
ciscoasa(config-if)# failover exec active router ospf 100
ciscoasa(config-if)#
```

Use the **show failover exec** command to display the command mode on the specified device in which commands sent with the **failover exec** command are executed. The **show failover exec** command takes the same keywords as the **failover exec** command: **active**, **mate**, or **standby**. The **failover exec** mode for each device is tracked separately.

For example, the following is sample output from the **show failover exec** command entered on the standby unit:

```
ciscoasa(config)# failover exec active interface GigabitEthernet0/1
ciscoasa(config)# sh failover exec active
Active unit Failover EXEC is at interface sub-command mode
```

```
ciscoasa(config)# sh failover exec standby
Standby unit Failover EXEC is at config mode

ciscoasa(config)# sh failover exec mate
Active unit Failover EXEC is at interface sub-command mode
```

Security Considerations

The **failover exec** command uses the failover link to send commands to and receive the output of the command execution from the peer unit. You should enable encryption on the failover link to prevent eavesdropping or man-in-the-middle attacks.

Limitations of Remote Command Execution

When you use remote commands, you face the following limitations:

- If you upgrade one unit using the zero-downtime upgrade procedure and not the other, both units must be running software that supports the **failover exec** command.
- Command completion and context help is not available for the commands in the *cmd_string* argument.
- In multiple context mode, you can only send commands to the peer context on the peer unit. To send commands to a different context, you must first change to that context on the unit to which you are logged in.
- You cannot use the following commands with the **failover exec** command:
  - **changeto**
  - **debug** (**undebug**)
- If the standby unit is in the failed state, it can still receive commands from the **failover exec** command if the failure is due to a service card failure; otherwise, the remote command execution will fail.
- You cannot use the **failover exec** command to switch from privileged EXEC mode to global configuration mode on the failover peer. For example, if the current unit is in privileged EXEC mode, and you enter **failover exec mate configure terminal**, the **show failover exec mate** output will show that the failover exec session is in global configuration mode. However, entering configuration commands for the peer unit using **failover exec** will fail until you enter global configuration mode on the current unit.
- You cannot enter recursive failover exec commands, such as the **failover exec mate failover exec mate** command.
- Commands that require user input or confirmation must use the **noconfirm** option. For example, to reload the mate, enter:

  failover exec mate reload noconfirm

Monitoring Failover

This section lets you monitor the Failover status.

Failover Messages

When a failover occurs, both ASAs send out system messages.

Failover Syslog Messages

The ASA issues a number of syslog messages related to failover at priority level 2, which indicates a critical condition. To view these messages, see the syslog messages guide. The ranges of message IDs associated with failover are: 101xxx, 102xxx, 103xxx, 104xxx, 105xxx, 210xxx, 311xxx, 709xxx, 727xxx. For example, 105032 and 105043 indicate a problem with the failover link.

Note During failover, the ASA logically shuts down and then brings up interfaces, generating syslog messages 411001 and 411002. This is normal activity.

Failover Debug Messages

To see debug messages, enter the **debug fover** command. See the command reference for more information.

Note Because debugging output is assigned high priority in the CPU process, it can drastically affect system performance. For this reason, use the **debug fover** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC.

SNMP Failover Traps

To receive SNMP syslog traps for failover, configure the SNMP agent to send SNMP traps to SNMP management stations, define a syslog host, and compile the Cisco syslog MIB into your SNMP management station.

Monitoring Failover Status

To monitor failover status, enter one of the following commands:

- **show failover**

  Displays information about the failover state of the unit.

- **show failover group**

  Displays information about the failover state of the failover group. The information displayed is similar to that of the **show failover** command but limited to the specified group.

- **show monitor-interface**

  Displays information about the monitored interface.

- **show running-config failover**

  Displays the failover commands in the running configuration.

History for Failover

Feature Name	Releases	Feature Information
Active/Standby failover	7.0(1)	This feature was introduced.
Active/Active failover	7.0(1)	This feature was introduced.
Support for a hex value for the failover key	7.0(4)	You can now specify a hex value for failover link encryption. We modified the following command: **failover key hex**.
Support for the master passphrase for the failover key	8.3(1)	The failover key now supports the master passphrase, which encrypts the shared key in the running and startup configuration. If you are copying the shared secret from one ASA to another, for example from the **more system:running-config** command, you can successfully copy and paste the encrypted shared key. **Note** The **failover key** shared secret shows as ***** in **show running-config** output; this obscured key is not copyable. We modified the following command: **failover key** [**0** | **8**].
IPv6 support for failover added.	8.2(2)	We modified the following commands: **failover interface ip**, **show failover**, **ipv6 address**, **show monitor-interface**.
Change to failover group unit preference during "simultaneous" bootup.	9.0(1)	Earlier software versions allowed “simultaneous” boot up so that the failover groups did not require the **preempt** command to become active on the preferred unit. However, this functionality has now changed so that both failover groups become active on the first unit to boot up.

Feature Name	Releases	Feature Information
Support for IPsec LAN-to-LAN tunnels to encrypt failover and state link communications	9.1(2)	Instead of using the proprietary encryption for the failover key (the **failover key** command), you can now use an IPsec LAN-to-LAN tunnel for failover and state link encryption. **Note** Failover LAN-to-LAN tunnels do not count against the IPsec (Other VPN) license. We introduced or modified the following commands: **failover ipsec pre-shared-key**, **show vpn-sessiondb**.
Disable health monitoring of a hardware module	9.3(1)	By default, the ASA monitors the health of an installed hardware module such as the ASA FirePOWER module. If you do not want a hardware module failure to trigger failover, you can disable module monitoring. We modified the following command: **monitor-interface service-module**
Lock configuration changes on the standby unit or standby context in a failover pair	9.3(2)	You can now lock configuration changes on the standby unit (Active/Standby failover) or the standby context (Active/Active failover) so you cannot make changes on the standby unit outside normal configuration syncing. We introduced the following command: **failover standby config-lock**

CHAPTER 9

ASA Cluster

Clustering lets you group multiple ASAs together as a single logical device. A cluster provides all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices.

Note Some features are not supported when using clustering. See Unsupported Features with Clustering, on page 339.

About ASA Clustering

This section describes the clustering architecture and how it works.

How the ASA Cluster Fits into Your Network

The cluster consists of multiple ASAs acting as a single unit. To act as a cluster, the ASAs need the following infrastructure:

- Isolated, high-speed backplane network for intra-cluster communication, known as the *cluster control link*.
- Management access to each ASA for configuration and monitoring.

When you place the cluster in your network, the upstream and downstream routers need to be able to load-balance the data coming to and from the cluster using one of the following methods:

- Spanned EtherChannel (Recommended)—Interfaces on multiple members of the cluster are grouped into a single EtherChannel; the EtherChannel performs load balancing between units.
- Policy-Based Routing (Routed firewall mode only)—The upstream and downstream routers perform load balancing between units using route maps and ACLs.
- Equal-Cost Multi-Path Routing (Routed firewall mode only)—The upstream and downstream routers perform load balancing between units using equal cost static or dynamic routes.

Cluster Members

Cluster members work together to accomplish the sharing of the security policy and traffic flows. This section describes the nature of each member role.

Bootstrap Configuration

On each device, you configure a minimal bootstrap configuration including the cluster name, cluster control link interface, and other cluster settings. The first unit on which you enable clustering typically becomes the *control* unit. When you enable clustering on subsequent units, they join the cluster as *data* units.

Control and Data Unit Roles

One member of the cluster is the control unit. The control unit is determined by the priority setting in the bootstrap configuration; the priority is set between 1 and 100, where 1 is the highest priority. All other members are data units. Typically, when you first create a cluster, the first unit you add becomes the control unit simply because it is the only unit in the cluster so far.

You must perform all configuration (aside from the bootstrap configuration) on the control unit only; the configuration is then replicated to the data units. In the case of physical assets, such as interfaces, the configuration of the control unit is mirrored on all data units. For example, if you configure GigabitEthernet 0/1 as the inside interface and GigabitEthernet 0/0 as the outside interface, then these interfaces are also used on the data units as inside and outside interfaces.

Some features do not scale in a cluster, and the control unit handles all traffic for those features.

Cluster Interfaces

You can configure data interfaces as either Spanned EtherChannels or as Individual interfaces. All data interfaces in the cluster must be one type only. See About Cluster Interfaces, on page 282 for more information.

Cluster Control Link

Each unit must dedicate at least one hardware interface as the cluster control link. See About the Cluster Control Link, on page 282 for more information.

Configuration Replication

All units in the cluster share a single configuration. You can only make configuration changes on the control unit, and changes are automatically synced to all other units in the cluster.

ASA Cluster Management

One of the benefits of using ASA clustering is the ease of management. This section describes how to manage the cluster.

Management Network

We recommend connecting all units to a single management network. This network is separate from the cluster control link.

Management Interface

For the management interface, we recommend using one of the dedicated management interfaces. You can configure the management interfaces as Individual interfaces (for both routed and transparent modes) or as a Spanned EtherChannel interface.

We recommend using Individual interfaces for management, even if you use Spanned EtherChannels for your data interfaces. Individual interfaces let you connect directly to each unit if necessary, while a Spanned EtherChannel interface only allows remote connection to the current control unit.

Note If you use Spanned EtherChannel interface mode, and configure the management interface as an Individual interface, you cannot enable dynamic routing for the management interface. You must use a static route.

For an Individual interface, the Main cluster IP address is a fixed address for the cluster that always belongs to the current control unit. For each interface, you also configure a range of addresses so that each unit, including the current control unit, can use a Local address from the range. The Main cluster IP address provides consistent management access to an address; when a control unit changes, the Main cluster IP address moves to the new control unit, so management of the cluster continues seamlessly. The Local IP address is used for routing, and is also useful for troubleshooting.

For example, you can manage the cluster by connecting to the Main cluster IP address, which is always attached to the current control unit. To manage an individual member, you can connect to the Local IP address.

For outbound management traffic such as TFTP or syslog, each unit, including the control unit, uses the Local IP address to connect to the server.

For a Spanned EtherChannel interface, you can only configure one IP address, and that IP address is always attached to the control unit. You cannot connect directly to a data unit using the EtherChannel interface; we recommend configuring the management interface as an Individual interface so that you can connect to each unit. Note that you can use a device-local EtherChannel for management.

Control Unit Management Vs. Data Unit Management

All management and monitoring can take place on the control unit. From the control unit, you can check runtime statistics, resource usage, or other monitoring information of all units. You can also issue a command to all units in the cluster, and replicate the console messages from data units to the control unit.

You can monitor data units directly if desired. Although also available from the control unit, you can perform file management on data units (including backing up the configuration and updating images). The following functions are not available from the control unit:

- Monitoring per-unit cluster-specific statistics.
- Syslog monitoring per unit (except for syslogs sent to the console when console replication is enabled).
- SNMP
- NetFlow

RSA Key Replication

When you create an RSA key on the control unit, the key is replicated to all data units. If you have an SSH session to the Main cluster IP address, you will be disconnected if the control unit fails. The new control unit uses the same key for SSH connections, so that you do not need to update the cached SSH host key when you reconnect to the new control unit.

ASDM Connection Certificate IP Address Mismatch

By default, a self-signed certificate is used for the ASDM connection based on the Local IP address. If you connect to the Main cluster IP address using ASDM, then a warning message about a mismatched IP address might appear because the certificate uses the Local IP address, and not the Main cluster IP address. You can ignore the message and establish the ASDM connection. However, to avoid this type of warning, you can enroll a certificate that contains the Main cluster IP address and all the Local IP addresses from the IP address pool. You can then use this certificate for each cluster member. See https://www.cisco.com/c/en/us/td/docs/security/asdm/identity-cert/cert-install.html for more information.

Inter-Site Clustering

For inter-site installations, you can take advantage of ASA clustering as long as you follow the recommended guidelines.

You can configure each cluster chassis to belong to a separate site ID.

Site IDs work with site-specific MAC addresses. Packets egressing the cluster use a site-specific MAC address, while packets received by the cluster use a global MAC address. This feature prevents the switches from learning the same global MAC address from both sites on two different ports, which causes MAC flapping; instead, they only learn the site MAC address. Site-specific MAC addresses are supported for routed mode using Spanned EtherChannels only.

See the following sections for more information about inter-site clustering:

- Sizing the Data Center Interconnect—Requirements and Prerequisites for ASA Clustering, on page 275
- Inter-Site Guidelines—Guidelines for ASA Clustering, on page 277
- Inter-Site Examples—Examples for Inter-Site Clustering, on page 336

Licenses for ASA Clustering

Cluster units do not require the same license on each unit. Typically, you buy a license only for the control unit; data units inherit the control unit license. If you have licenses on multiple units, they combine into a single running ASA cluster license.

There are exceptions to this rule. See the following table for precise licensing requirements for clustering.

Model	License Requirement
ASA 5585-X	Cluster License, supports up to 16 units. **Note** Each unit must have the same encryption license; each unit must have the same 10 GE I/O/Security Plus license (ASA 5585-X with SSP-10 and -20).
ASA 5512-X	Security Plus license, supports 2 units. **Note** Each unit must have the same encryption license.
ASA 5515-X, ASA 5525-X, ASA 5545-X, ASA 5555-X	Base License, supports 2 units. **Note** Each unit must have the same encryption license.
Firepower 9300 Chassis	See ASA Cluster Licenses for the ASA on the Firepower 9300 Chassis, on page 114.
All other models	No support.

Requirements and Prerequisites for ASA Clustering

Model Requirements

- ASA 5512-X, 5515-X, 5525-X, 5545-X, and 5555-X—Maximum 2 units
- ASA 5585-X—Maximum 16 units

  For the ASA 5585-X with SSP-10 and SSP-20, which include two Ten Gigabit Ethernet interfaces, we recommend using one interface for the cluster control link, and the other for data (you can use subinterfaces for data). Although this setup does not accommodate redundancy for the cluster control link, it does satisfy the need to size the cluster control link to match the size of the data interfaces.
- ASA FirePOWER module—The ASA FirePOWER module does not support clustering directly, but you can use these modules in a cluster. You are responsible for maintaining consistent policies on the ASA FirePOWER modules in the cluster.

Note Create the cluster before you configure the ASA FirePOWER modules. If the modules are already configured on the data units, clear the interface configuration on the devices before adding them to the cluster. From the CLI, enter the **clear configure interface** command.

ASA Hardware and Software Requirements

All units in a cluster:

- Must be the same model with the same DRAM. You do not have to have the same amount of flash memory.
- Must run the identical software except at the time of an image upgrade. Hitless upgrade is supported.
- Must be in the same security context mode, single or multiple.
- (Single context mode) Must be in the same firewall mode, routed or transparent.
- New cluster members must use the same SSL encryption setting (the **ssl encryption** command) as the control unit for initial cluster control link communication before configuration replication.
- Must have the same cluster, encryption and, for the ASA 5585-X, 10 GE I/O licenses.

Switch Requirements

- Be sure to complete the switch configuration before you configure clustering on the ASAs.
- For a list of supported switches, see Cisco ASA Compatibility.

ASA Requirements

- Provide each unit with a unique IP address before you join them to the management network.
  - See the Getting Started chapter for more information about connecting to the ASA and setting the management IP address.
  - Except for the IP address used by the control unit (typically the first unit you add to the cluster), these management IP addresses are for temporary use only.
  - After a data unit joins the cluster, its management interface configuration is replaced by the one replicated from the control unit.
- To use jumbo frames on the cluster control link (recommended), you must enable Jumbo Frame Reservation before you enable clustering.

Sizing the Data Center Interconnect for Inter-Site Clustering

You should reserve bandwidth on the data center interconnect (DCI) for cluster control link traffic equivalent to the following calculation:

$$\frac{\text{\# of cluster members per site}}{2} \times \text{cluster control link size per member}$$

If the number of members differs at each site, use the larger number for your calculation. The minimum bandwidth for the DCI should not be less than the size of the cluster control link for one member.

For example:

- For 4 members at 2 sites:
  - 4 cluster members total
  - 2 members at each site
  - 5 Gbps cluster control link per member

  Reserved DCI bandwidth = 5 Gbps (2/2 x 5 Gbps).
- For 6 members at 3 sites, the size increases:
  - 6 cluster members total
  - 3 members at site 1, 2 members at site 2, and 1 member at site 3
  - 10 Gbps cluster control link per member

  Reserved DCI bandwidth = 15 Gbps (3/2 x 10 Gbps).
- For 2 members at 2 sites:
  - 2 cluster members total
  - 1 member at each site
  - 10 Gbps cluster control link per member

  Reserved DCI bandwidth = 10 Gbps (1/2 x 10 Gbps = 5 Gbps; but the minimum bandwidth should not be less than the size of the cluster control link (10 Gbps)).

Other Requirements

We recommend using a terminal server to access all cluster member unit console ports. For initial setup, and ongoing management (for example, when a unit goes down), a terminal server is useful for remote management.

Guidelines for ASA Clustering

Context Mode

The mode must match on each member unit.

Firewall Mode

For single mode, the firewall mode must match on all units.

Failover

Failover is not supported with clustering.

IPv6

The cluster control link is only supported using IPv4.

Switches

- Make sure connected switches match the MTU for both cluster data interfaces and the cluster control link interface. You should configure the cluster control link interface MTU to be at least 100 bytes higher than the data interface MTU, so make sure to configure the cluster control link connecting switch appropriately. Because the cluster control link traffic includes data packet forwarding, the cluster control link needs to accommodate the entire size of a data packet plus cluster traffic overhead.
- For Cisco IOS XR systems, if you want to set a non-default MTU, set the IOS interface MTU to be 14 bytes higher than the cluster device MTU. Otherwise, OSPF adjacency peering attempts may fail unless the **mtu-ignore** option is used. Note that the cluster device MTU should match the IOS *IPv4* MTU. This adjustment is not required for Cisco Catalyst and Cisco Nexus switches.
- On the switch(es) for the cluster control link interfaces, you can optionally enable Spanning Tree PortFast on the switch ports connected to the cluster unit to speed up the join process for new units.
- On the switch, we recommend that you use one of the following EtherChannel load-balancing algorithms: **source-dest-ip** or **source-dest-ip-port** (see the Cisco Nexus OS and Cisco IOS **port-channel load-balance** command). Do not use a **vlan** keyword in the load-balance algorithm because it can cause unevenly distributed traffic to the devices in a cluster. *Do not* change the load-balancing algorithm from the default on the cluster device.
- If you change the load-balancing algorithm of the EtherChannel on the switch, the EtherChannel interface on the switch temporarily stops forwarding traffic, and the Spanning Tree Protocol restarts. There will be a delay before traffic starts flowing again.
- Some switches do not support dynamic port priority with LACP (active and standby links). You can disable dynamic port priority to provide better compatibility with Spanned EtherChannels.
- Switches on the cluster control link path should not verify the L4 checksum. Redirected traffic over the cluster control link does not have a correct L4 checksum. Switches that verify the L4 checksum could cause traffic to be dropped.
- Port-channel bundling downtime should not exceed the configured keepalive interval.
- On Supervisor 2T EtherChannels, the default hash distribution algorithm is adaptive. To avoid asymmetric traffic in a VSS design, change the hash algorithm on the port-channel connected to the cluster device to fixed:

  router(config)# **port-channel** *id* **hash-distribution fixed**

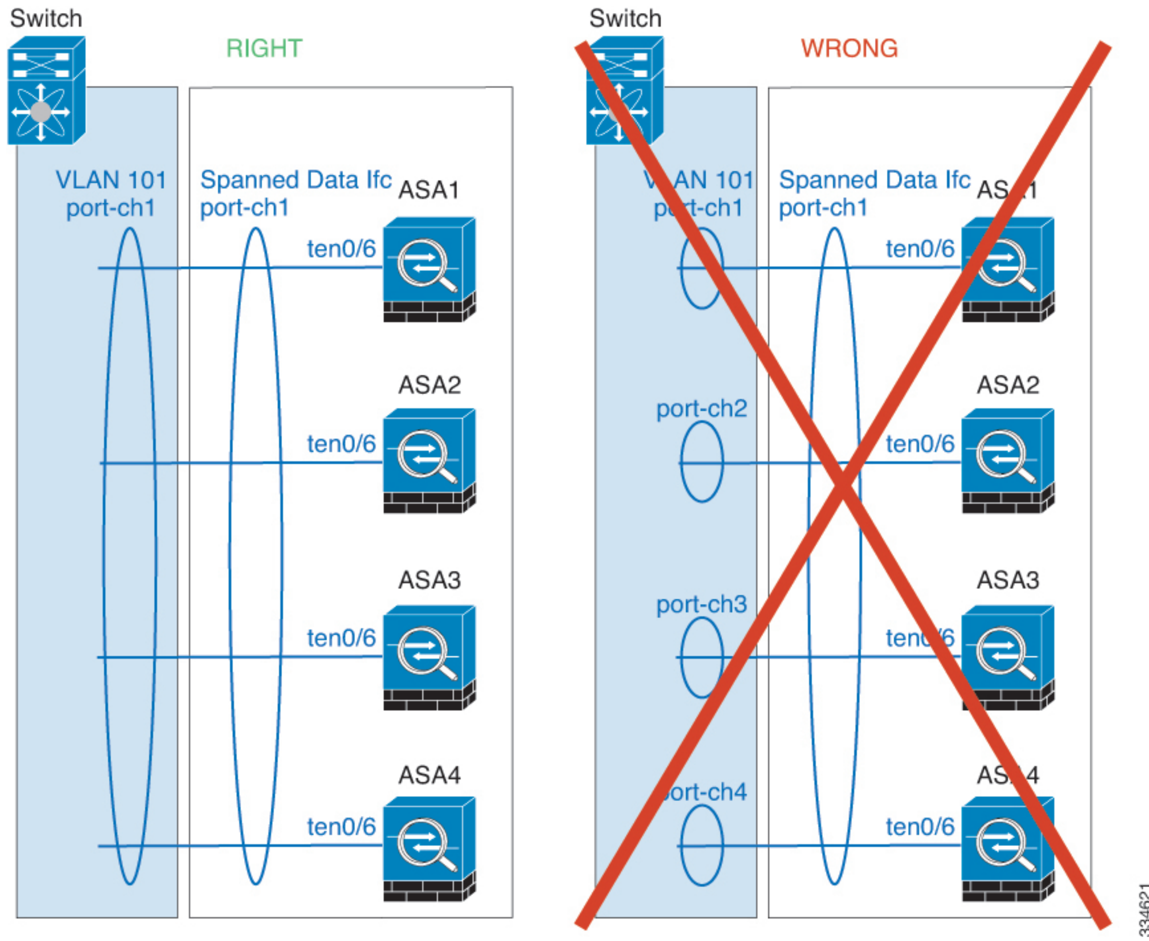
  Do not change the algorithm globally; you may want to take advantage of the adaptive algorithm for the VSS peer link.
- You should disable the LACP Graceful Convergence feature on all cluster-facing EtherChannel interfaces for Cisco Nexus switches.

EtherChannels

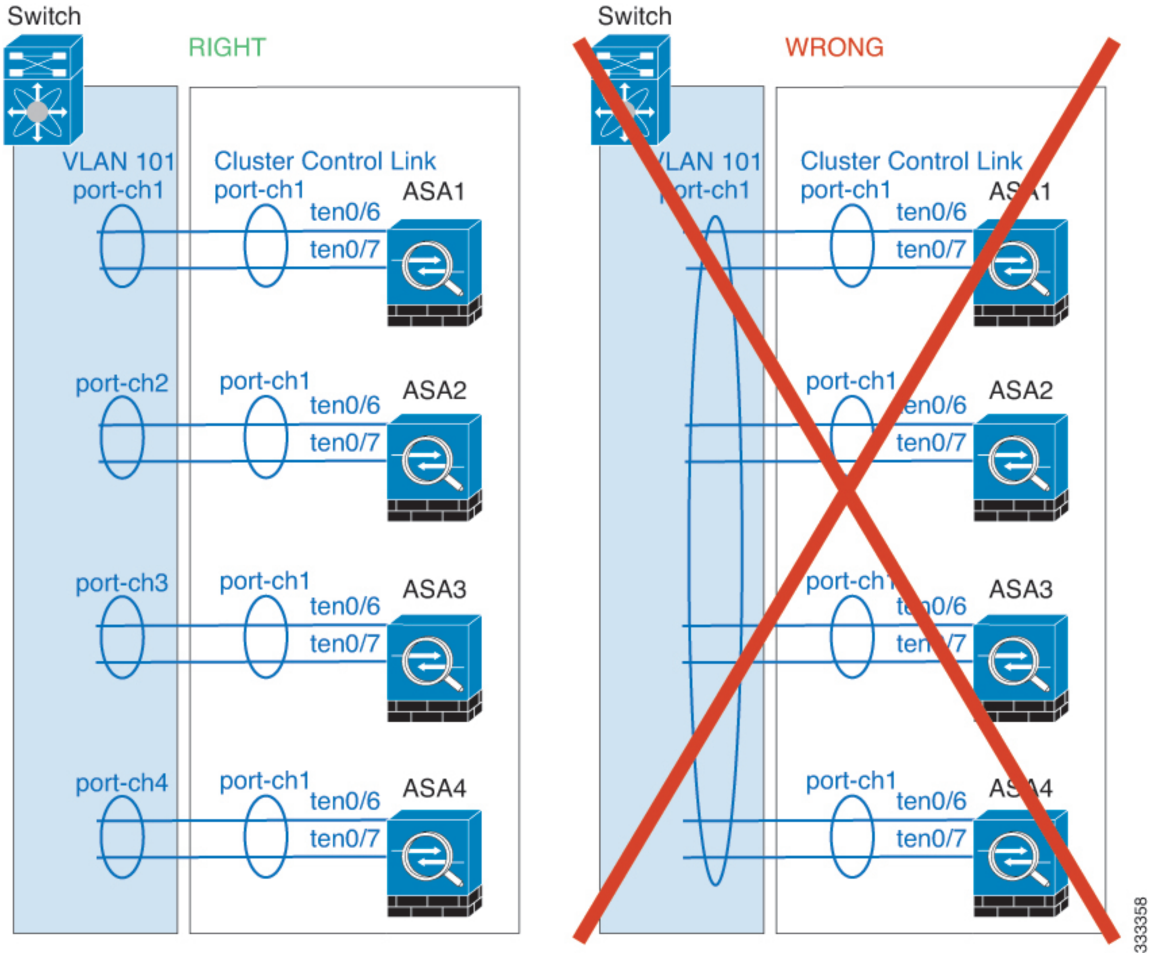
- In Catalyst 3750-X Cisco IOS software versions earlier than 15.1(1)S2, the cluster unit did not support connecting an EtherChannel to a switch stack. With default switch settings, if the cluster unit EtherChannel is connected cross stack, and if the control unit switch is powered down, then the EtherChannel connected to the remaining switch will not come up. To improve compatibility, set the **stack-mac persistent timer** command to a large enough value to account for reload time; for example, 8 minutes or 0 for indefinite. Or, you can upgrade to more a more stable switch software version, such as 15.1(1)S2.

- Spanned vs. Device-Local EtherChannel Configuration—Be sure to configure the switch appropriately for Spanned EtherChannels vs. Device-local EtherChannels.

  - Spanned EtherChannels—For cluster unit *Spanned* EtherChannels, which span across all members of the cluster, the interfaces are combined into a single EtherChannel on the switch. Make sure each interface is in the same channel group on the switch.



  - Device-local EtherChannels—For cluster unit *Device-local* EtherChannels including any EtherChannels configured for the cluster control link, be sure to configure discrete EtherChannels on the switch; do not combine multiple cluster unit EtherChannels into one EtherChannel on the switch.



Inter-Site Guidelines

See the following guidelines for inter-site clustering:

- Supports inter-site clustering in the following interface and firewall modes:

Interface Mode	Firewall Mode	
	Routed	**Transparent**
Individual Interface	**Yes**	N/A
Spanned EtherChannel	No	**Yes**

- For individual interface mode, when using ECMP towards a multicast Rendezvous Point (RP), we recommend that you use a static route for the RP IP address using the Main cluster IP address as the next hop. This static route prevents sending unicast PIM register packets to data units. If a data unit receives a PIM register packet, then the packet is dropped, and the multicast stream cannot be registered.

- The cluster control link latency must be less than 20 ms round-trip time (RTT).

- The cluster control link must be reliable, with no out-of-order or dropped packets; for example, you should use a dedicated link.
- Do not configure connection rebalancing; you do not want connections rebalanced to cluster members at a different site.
- The ASA does not encrypt forwarded data traffic on the cluster control link because it is a dedicated link, even when used on a Data Center Interconnect (DCI). If you use Overlay Transport Virtualization (OTV), or are otherwise extending the cluster control link outside of the local administrative domain, you can configure encryption on your border routers such as 802.1AE MacSec over OTV.
- The cluster implementation does not differentiate between members at multiple sites for incoming connections; therefore, connection roles for a given connection may span across sites. This is expected behavior.
- For transparent mode, if the cluster is placed between a pair of inside and outside routers (AKA North-South insertion), you must ensure that both inside routers share a MAC address, and also that both outside routers share a MAC address. When a cluster member at site 1 forwards a connection to a member at site 2, the destination MAC address is preserved. The packet will only reach the router at site 2 if the MAC address is the same as the router at site 1.
- For transparent mode, if the cluster is placed between data networks and the gateway router at each site for firewalling between internal networks (AKA East-West insertion), then each gateway router should use a First Hop Redundancy Protocol (FHRP) such as HSRP to provide identical virtual IP and MAC address destinations at each site. The data VLANs are extended across the sites using Overlay Transport Virtualization (OTV), or something similar. You need to create filters to prevent traffic that is destined to the local gateway router from being sent over the DCI to the other site. If the gateway router becomes unreachable at one site, you need to remove any filters so traffic can successfully reach the other site's gateway.
- For transparent mode, if the cluster is connected to an HSRP router, you must add the router HSRP MAC address as a static MAC address table entry on the ASA (see Add a Static MAC Address for Bridge Groups, on page 591). When adjacent routers use HSRP, traffic destined to the HSRP IP address will be sent to the HSRP MAC Address, but return traffic will be sourced from the MAC address of a particular router's interface in the HSRP pair. Therefore, the ASA MAC address table is typically only updated when the ASA ARP table entry for the HSRP IP address expires, and the ASA sends an ARP request and receives a reply. Because the ASA's ARP table entries expire after 14400 seconds by default, but the MAC address table entry expires after 300 seconds by default, a static MAC address entry is required to avoid MAC address table expiration traffic drops.

Additional Guidelines

- When significant topology changes occur (such as adding or removing an EtherChannel interface, enabling or disabling an interface on the ASA or the switch, adding an additional switch to form a VSS or vPC) you should disable the health check feature and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all units, you can re-enable the interface health check feature.
- When adding a unit to an existing cluster, or when reloading a unit, there will be a temporary, limited packet/connection drop; this is expected behavior. In some cases, the dropped packets can hang your connection; for example, dropping a FIN/ACK packet for an FTP connection will make the FTP client hang. In this case, you need to reestablish the FTP connection.

- If you use a Windows 2003 server connected to a Spanned EtherChannel, when the syslog server port is down and the server does not throttle ICMP error messages, then large numbers of ICMP messages are sent back to the ASA cluster. These messages can result in some units of the ASA cluster experiencing high CPU, which can affect performance. We recommend that you throttle ICMP error messages.
- We do not support VXLAN in Individual Interface mode. Only Spanned EtherChannel mode supports VXLAN.
- It takes time to replicate changes to all the units in a cluster. If you make a large change, for example, adding an access control rule that uses object groups (which, when deployed, are broken out into multiple rules), the time needed to complete the change can exceed the timeout for the cluster units to respond with a success message. If this happens, you might see a "failed to replicate command" message. You can ignore the message.

Defaults for ASA Clustering

- When using Spanned EtherChannels, the cLACP system ID is auto-generated and the system priority is 1 by default.
- The cluster health check feature is enabled by default with the holdtime of 3 seconds. Interface health monitoring is enabled on all interfaces by default.
- Connection rebalancing is disabled by default. If you enable connection rebalancing, the default time between load information exchanges is 5 seconds.

Configure ASA Clustering

To configure clustering, perform the following tasks.

Note To enable or disable clustering, you must use a console connection (for CLI) or an ASDM connection.

Cable the Units and Configure Interfaces

Before configuring clustering, cable the cluster control link network, management network, and data networks. Then configure your interfaces.

About Cluster Interfaces

You can configure data interfaces as either Spanned EtherChannels or as Individual interfaces. All data interfaces in the cluster must be one type only. Each unit must also dedicate at least one hardware interface as the cluster control link.

About the Cluster Control Link

Each unit must dedicate at least one hardware interface as the cluster control link.

Cluster Control Link Traffic Overview

Cluster control link traffic includes both control and data traffic.

Control traffic includes:

- Control unit election.
- Configuration replication.
- Health monitoring.

Data traffic includes:

- State replication.
- Connection ownership queries and data packet forwarding.

Cluster Control Link Interfaces and Network

You can use any data interface(s) for the cluster control link, with the following exceptions:

- You cannot use a VLAN subinterface as the cluster control link.
- You cannot use a Management *x/x* interface as the cluster control link, either alone or as an EtherChannel.
- For the ASA 5585-X with an ASA FirePOWER module, Cisco recommends that you use ASA interfaces for the cluster control link, and not interfaces on the ASA FirePOWER module. Module interfaces can drop traffic for up to 30 seconds during a module reload, including reloads that occur during a software upgrade. However, if needed, you can use module interfaces and ASA interfaces in the same cluster control link EtherChannel. When the module interfaces drop, the remaining interfaces in the EtherChannel are still up. The ASA 5585-X Network Module does not run a separate operating system, so it is not affected by this issue.

  Be aware that data interfaces on the module are also affected by reload drops. Cisco recommends always using ASA interfaces redundantly with module interfaces in an EtherChannel.

  For the ASA 5585-X with SSP-10 and SSP-20, which include two Ten Gigabit Ethernet interfaces, we recommend using one interface for the cluster control link, and the other for data (you can use subinterfaces for data). Although this setup does not accommodate redundancy for the cluster control link, it does satisfy the need to size the cluster control link to match the size of the data interfaces.

You can use an EtherChannel or redundant interface.

Each cluster control link has an IP address on the same subnet. This subnet should be isolated from all other traffic, and should include only the ASA cluster control link interfaces.

For a 2-member cluster, do not directly-connect the cluster control link from one ASA to the other ASA. If you directly connect the interfaces, then when one unit fails, the cluster control link fails, and thus the remaining healthy unit fails. If you connect the cluster control link through a switch, then the cluster control link remains up for the healthy unit.

Size the Cluster Control Link

If possible, you should size the cluster control link to match the expected throughput of each chassis so the cluster-control link can handle the worst-case scenarios. For example, if you have the ASA 5585-X with SSP-60, which can pass 14 Gbps per unit maximum in a cluster, then you should also assign interfaces to the cluster control link that can pass at least 14 Gbps. In this case, you could use 2 Ten Gigabit Ethernet interfaces in an EtherChannel for the cluster control link, and use the rest of the interfaces as desired for data links.

Cluster control link traffic is comprised mainly of state update and forwarded packets. The amount of traffic at any given time on the cluster control link varies. The amount of forwarded traffic depends on the load-balancing efficacy or whether there is a lot of traffic for centralized features. For example:

- NAT results in poor load balancing of connections, and the need to rebalance all returning traffic to the correct units.
- AAA for network access is a centralized feature, so all traffic is forwarded to the control unit.
- When membership changes, the cluster needs to rebalance a large number of connections, thus temporarily using a large amount of cluster control link bandwidth.

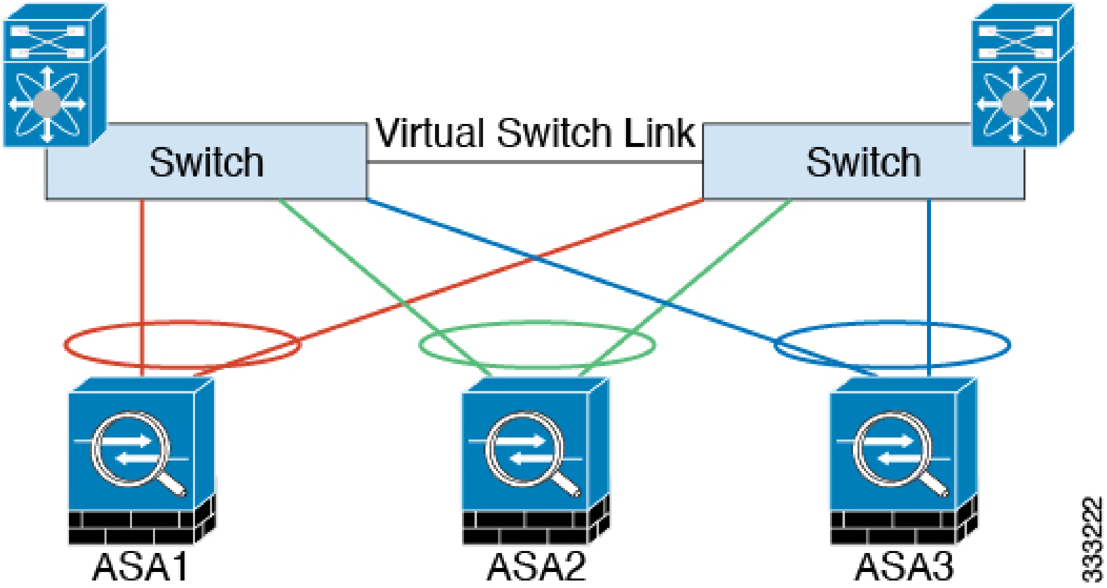
A higher-bandwidth cluster control link helps the cluster to converge faster when there are membership changes and prevents throughput bottlenecks.

Note If your cluster has large amounts of asymmetric (rebalanced) traffic, then you should increase the cluster control link size.

Cluster Control Link Redundancy

We recommend using an EtherChannel for the cluster control link, so that you can pass traffic on multiple links in the EtherChannel while still achieving redundancy.

The following diagram shows how to use an EtherChannel as a cluster control link in a Virtual Switching System (VSS) or Virtual Port Channel (vPC) environment. All links in the EtherChannel are active. When the switch is part of a VSS or vPC, then you can connect ASA interfaces within the same EtherChannel to separate switches in the VSS or vPC. The switch interfaces are members of the same EtherChannel port-channel interface, because the separate switches act like a single switch. Note that this EtherChannel is device-local, not a Spanned EtherChannel.



Cluster Control Link Reliability

To ensure cluster control link functionality, be sure the round-trip time (RTT) between units is less than 20 ms. This maximum latency enhances compatibility with cluster members installed at different geographical sites. To check your latency, perform a ping on the cluster control link between units.

The cluster control link must be reliable, with no out-of-order or dropped packets; for example, for inter-site deployment, you should use a dedicated link.

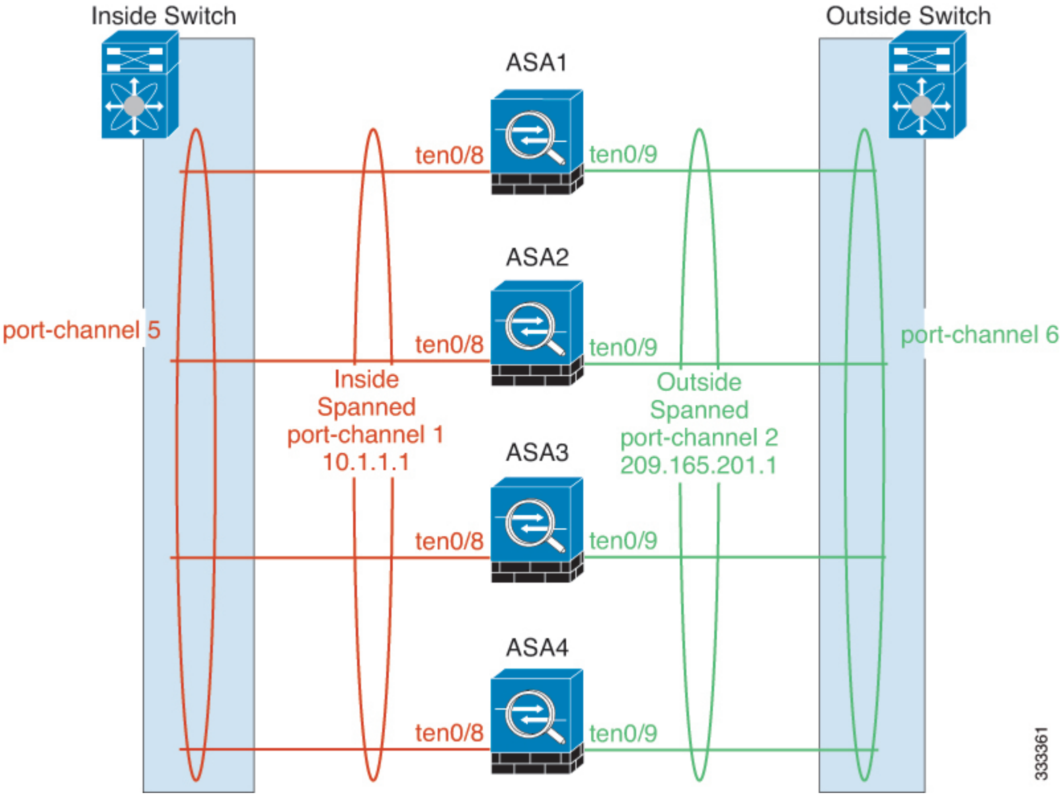
Cluster Control Link Failure

If the cluster control link line protocol goes down for a unit, then clustering is disabled; data interfaces are shut down. After you fix the cluster control link, you must manually rejoin the cluster by re-enabling clustering.

Note When the ASA becomes inactive, all data interfaces are shut down; only the management-only interface can send and receive traffic. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster, the management interface is not accessible (because it then uses the Main IP address, which is the same as the control unit). You must use the console port for any further configuration.

Spanned EtherChannels (Recommended)

You can group one or more interfaces per chassis into an EtherChannel that spans all chassis in the cluster. The EtherChannel aggregates the traffic across all the available active interfaces in the channel. A Spanned EtherChannel can be configured in both routed and transparent firewall modes. In routed mode, the EtherChannel is configured as a routed interface with a single IP address. In transparent mode, the IP address is assigned to the BVI, not to the bridge group member interface. The EtherChannel inherently provides load balancing as part of basic operation.



Spanned EtherChannel Benefits

The EtherChannel method of load-balancing is recommended over other methods for the following benefits:

- Faster failure discovery.
- Faster convergence time. Individual interfaces rely on routing protocols to load-balance traffic, and routing protocols often have slow convergence during a link failure.
- Ease of configuration.

Guidelines for Maximum Throughput

To achieve maximum throughput, we recommend the following:

- Use a load balancing hash algorithm that is “symmetric,” meaning that packets from both directions will have the same hash, and will be sent to the same ASA in the Spanned EtherChannel. We recommend using the source and destination IP address (the default) or the source and destination port as the hashing algorithm.
- Use the same type of line cards when connecting the ASAs to the switch so that hashing algorithms applied to all packets are the same.

Load Balancing

The EtherChannel link is selected using a proprietary hash algorithm, based on source or destination IP addresses and TCP and UDP port numbers.

Note On the ASA, do not change the load-balancing algorithm from the default. On the switch, we recommend that you use one of the following algorithms: **source-dest-ip** or **source-dest-ip-port** (see the Cisco Nexus OS or Cisco IOS **port-channel load-balance** command). Do not use a **vlan** keyword in the load-balance algorithm because it can cause unevenly distributed traffic to the ASAs in a cluster.

The number of links in the EtherChannel affects load balancing.

Symmetric load balancing is not always possible. If you configure NAT, then forward and return packets will have different IP addresses and/or ports. Return traffic will be sent to a different unit based on the hash, and the cluster will have to redirect most returning traffic to the correct unit.

EtherChannel Redundancy

The EtherChannel has built-in redundancy. It monitors the line protocol status of all links. If one link fails, traffic is re-balanced between remaining links. If all links in the EtherChannel fail on a particular unit, but other units are still active, then the unit is removed from the cluster.

Connecting to a VSS or vPC

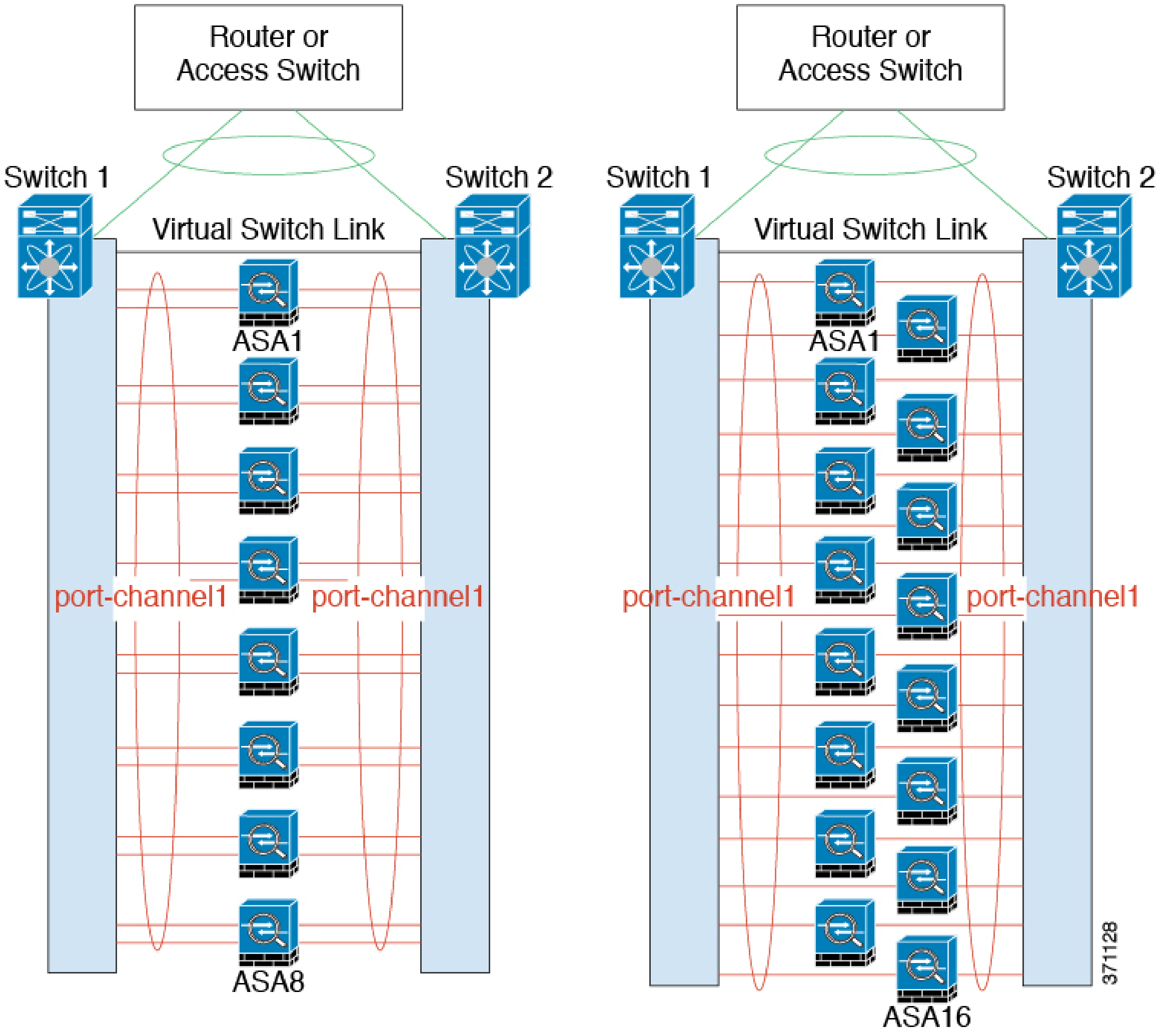
You can include multiple interfaces per ASA in the Spanned EtherChannel. Multiple interfaces per ASA are especially useful for connecting to both switches in a VSS or vPC.

Depending on your switches, you can configure up to 32 active links in the spanned EtherChannel. This feature requires both switches in the vPC to support EtherChannels with 16 active links each (for example the Cisco Nexus 7000 with F2-Series 10 Gigabit Ethernet Module).

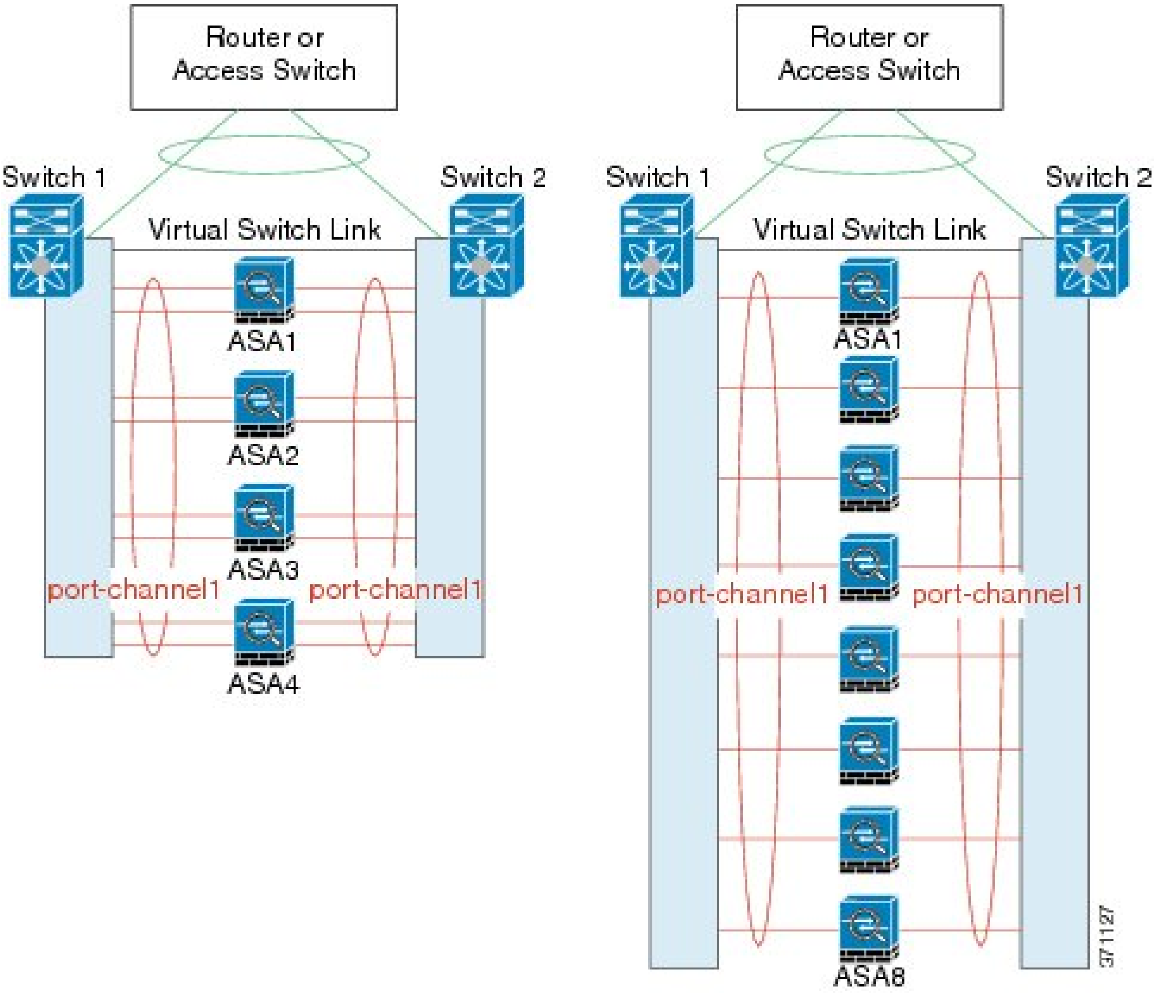
For switches that support 8 active links in the EtherChannel, you can configure up to 16 active links in the spanned EtherChannel when connecting to two switches in a VSS/vPC.

If you want to use more than 8 active links in a spanned EtherChannel, you cannot also have standby links; the support for 9 to 32 active links requires you to disable cLACP dynamic port priority that allows the use of standby links. You can still use 8 active links and 8 standby links if desired, for example, when connecting to a single switch.

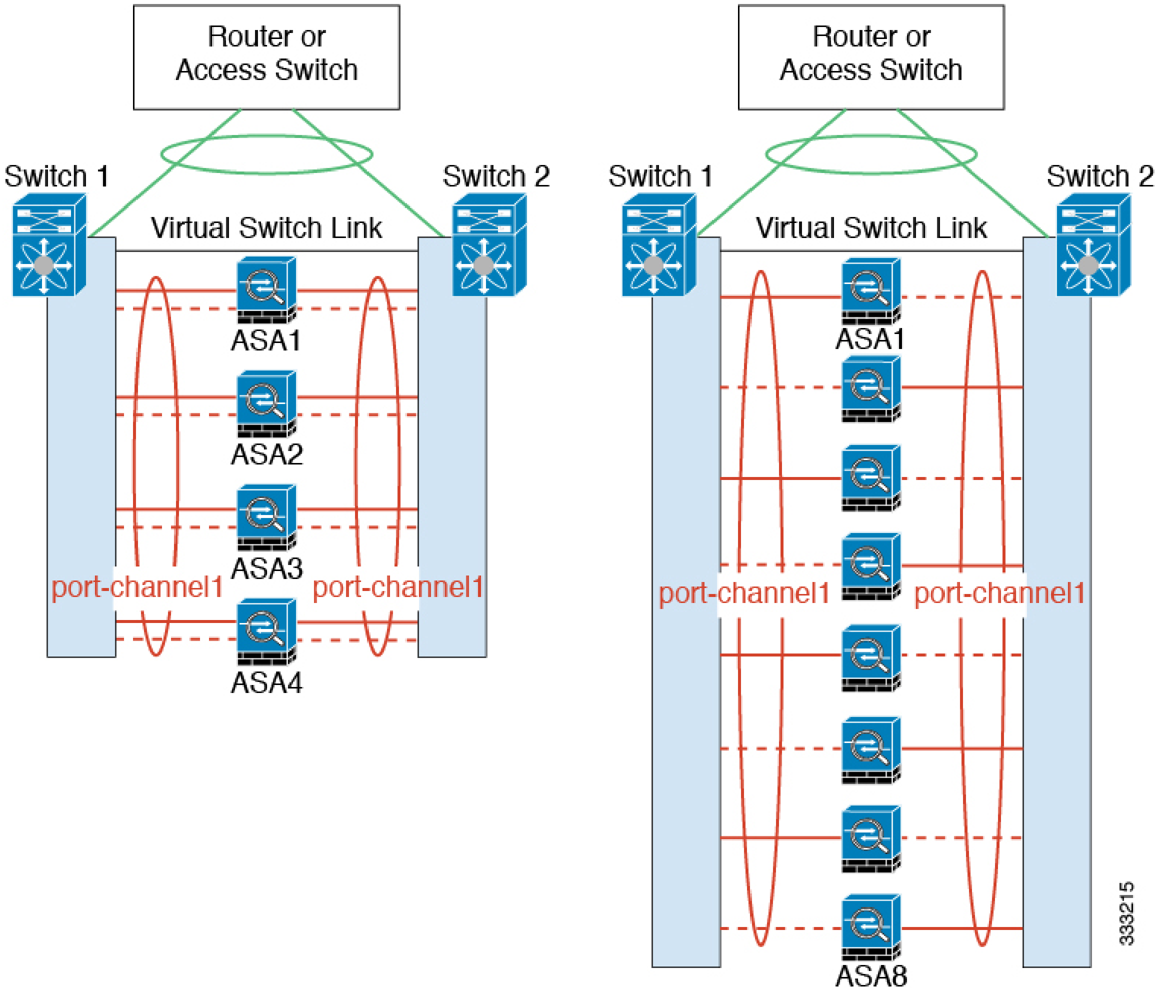
The following figure shows a 32 active link spanned EtherChannel in an 8-ASA cluster and a 16-ASA cluster.



The following figure shows a 16 active link spanned EtherChannel in a 4-ASA cluster and an 8-ASA cluster.



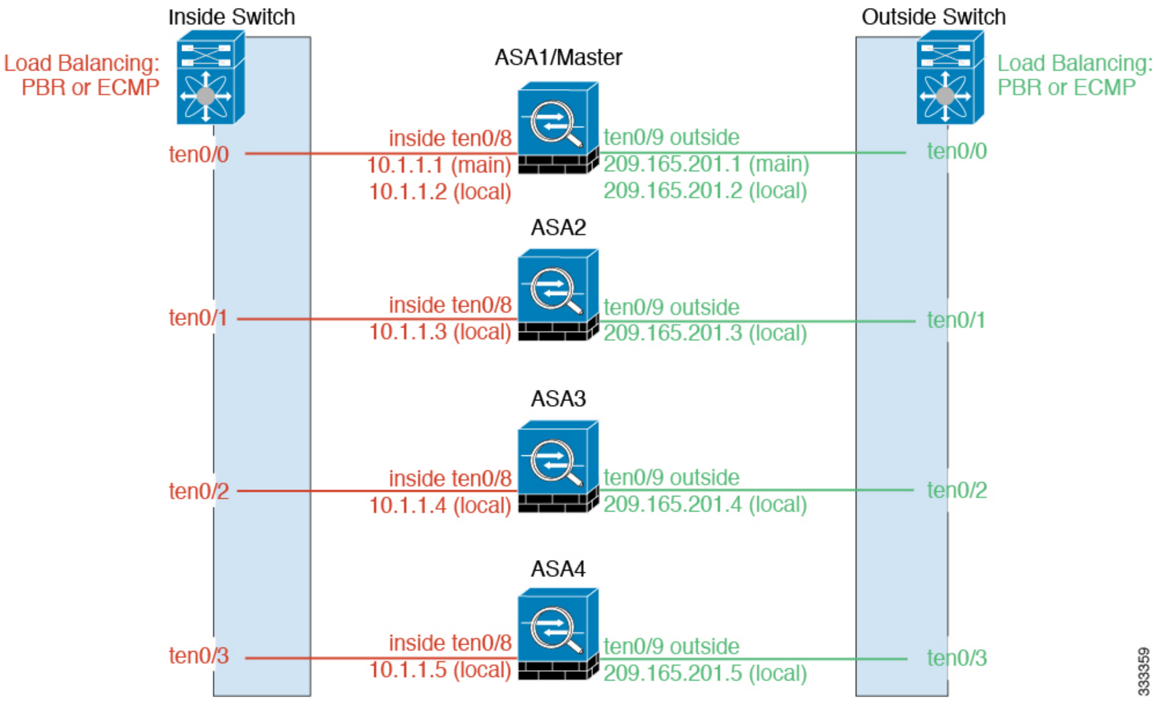
The following figure shows a traditional 8 active/8 standby link spanned EtherChannel in a 4-ASA cluster and an 8-ASA cluster. The active links are shown as solid lines, while the inactive links are dotted. cLACP load-balancing can automatically choose the best 8 links to be active in the EtherChannel. As shown, cLACP helps achieve load balancing at the link level.



Individual Interfaces (Routed Firewall Mode Only)

Individual interfaces are normal routed interfaces, each with their own *Local IP address*. Because interface configuration must be configured only on the control unit, the interface configuration lets you set a pool of IP addresses to be used for a given interface on the cluster members, including one for the control unit. The *Main cluster IP address* is a fixed address for the cluster that always belongs to the current control unit. The Main cluster IP address is a data unit IP address for the control unit; the Local IP address is always the control unit address for routing. The Main cluster IP address provides consistent management access to an address; when a control unit changes, the Main cluster IP address moves to the new control unit, so management of the cluster continues seamlessly. Load balancing, however, must be configured separately on the upstream switch in this case.

Note We recommend Spanned EtherChannels instead of Individual interfaces because Individual interfaces rely on routing protocols to load-balance traffic, and routing protocols often have slow convergence during a link failure.



Policy-Based Routing (Routed Firewall Mode Only)

When using Individual interfaces, each ASA interface maintains its own IP address and MAC address. One method of load balancing is Policy-Based Routing (PBR).

We recommend this method if you are already using PBR, and want to take advantage of your existing infrastructure. This method might offer additional tuning options vs. Spanned EtherChannel as well.

PBR makes routing decisions based on a route map and ACL. You must manually divide traffic between all ASAs in a cluster. Because PBR is static, it may not achieve the optimum load balancing result at all times. To achieve the best performance, we recommend that you configure the PBR policy so that forward and return packets of a connection are directed to the same physical ASA. For example, if you have a Cisco router, redundancy can be achieved by using Cisco IOS PBR with Object Tracking. Cisco IOS Object Tracking monitors each ASA using ICMP ping. PBR can then enable or disable route maps based on reachability of a particular ASA. See the following URLs for more details:

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/intelligent-traffic-director/index.html

http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a4409.shtml

Note If you use this method of load-balancing, you can use a device-local EtherChannel as an Individual interface.

Equal-Cost Multi-Path Routing (Routed Firewall Mode Only)

When using Individual interfaces, each ASA interface maintains its own IP address and MAC address. One method of load balancing is Equal-Cost Multi-Path (ECMP) routing.

We recommend this method if you are already using ECMP, and want to take advantage of your existing infrastructure. This method might offer additional tuning options vs. Spanned EtherChannel as well.

ECMP routing can forward packets over multiple “best paths” that tie for top place in the routing metric. Like EtherChannel, a hash of source and destination IP addresses and/or source and destination ports can be used to send a packet to one of the next hops. If you use static routes for ECMP routing, then an ASA failure can cause problems; the route continues to be used, and traffic to the failed ASA will be lost. If you use static routes, be sure to use a static route monitoring feature such as Object Tracking. We recommend using dynamic routing protocols to add and remove routes, in which case, you must configure each ASA to participate in dynamic routing.

Note If you use this method of load-balancing, you can use a device-local EtherChannel as an Individual interface.

Nexus Intelligent Traffic Director (Routed Firewall Mode Only)

When using Individual interfaces, each ASA interface maintains its own IP address and MAC address. Intelligent Traffic Director (ITD) is a high-speed hardware load-balancing solution for Nexus 5000, 6000, 7000, and 9000 switch series. In addition to fully covering the functional capabilities of traditional PBR, it offers a simplified configuration workflow and multiple additional features for a more granular load distribution.

ITD supports IP stickiness, consistent hashing for bi-directional flow symmetry, virtual IP addressing, health monitoring, sophisticated failure handling policies with N+M redundancy, weighted load-balancing, and application IP SLA probes including DNS. Due to the dynamic nature of load-balancing, it achieves a more even traffic distribution across all cluster members as compared to PBR. In order to achieve bi-directional flow symmetry, we recommend configuring ITD such that forward and return packets of a connection are directed to the same physical ASA. See the following URL for more details:

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/intelligent-traffic-director/index.html

Cable the Cluster Units and Configure Upstream and Downstream Equipment

Before configuring clustering, cable the cluster control link network, management network, and data networks.

Procedure

Cable the cluster control link network, management network, and data networks.

Note At a minimum, an active cluster control link network is required before you configure the units to join the cluster.

You should also configure the upstream and downstream equipment. For example, if you use EtherChannels, then you should configure the upstream and downstream equipment for the EtherChannels.

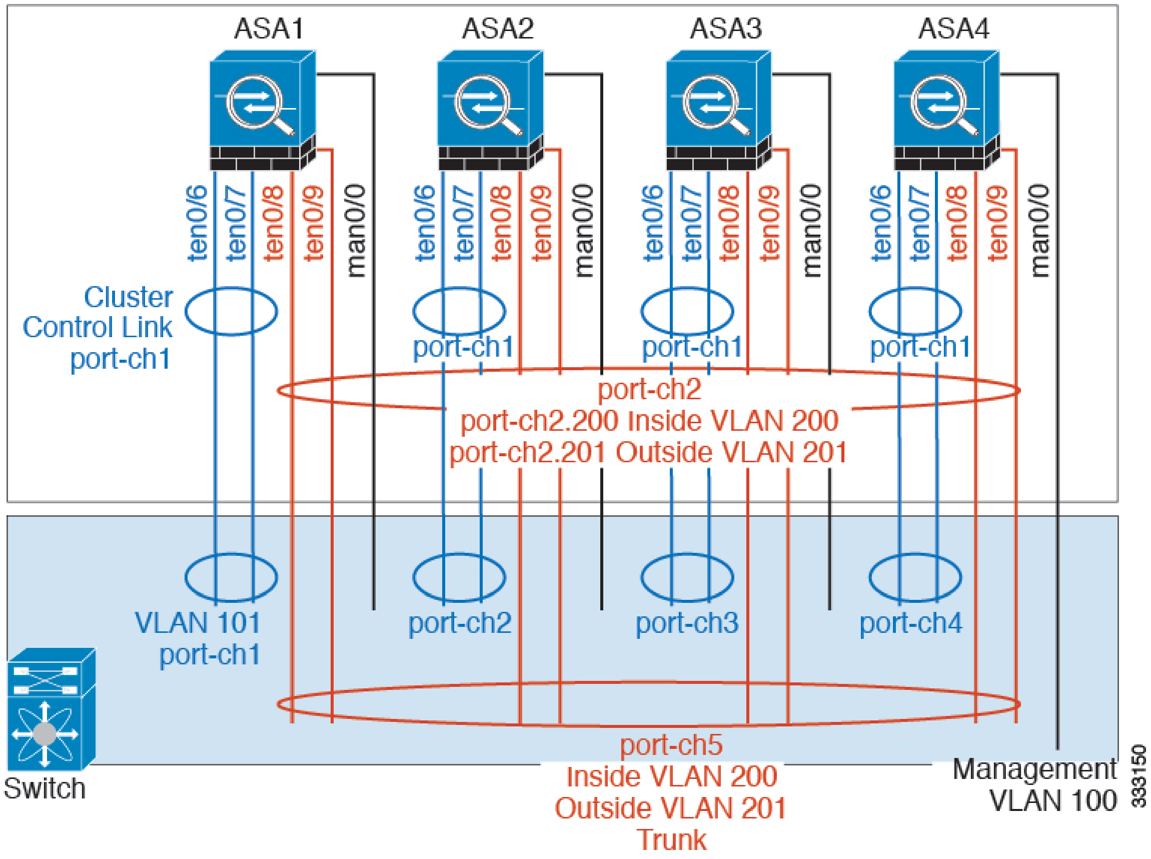
Examples

Note This example uses EtherChannels for load-balancing. If you are using PBR or ECMP, your switch configuration will differ.

For example on each of 4 ASA 5585-Xs, you want to use:

- 2 Ten Gigabit Ethernet interfaces in a device-local EtherChannel for the cluster control link.
- 2 Ten Gigabit Ethernet interfaces in a Spanned EtherChannel for the inside and outside network; each interface is a VLAN subinterface of the EtherChannel. Using subinterfaces lets both inside and outside interfaces take advantage of the benefits of an EtherChannel.
- 1 Management interface.

You have one switch for both the inside and outside networks.



Purpose	Connect Interfaces on each of 4 ASAs	To Switch Ports
Cluster control link	TenGigabitEthernet 0/6 and TenGigabitEthernet 0/7	8 ports total For each TenGigabitEthernet 0/6 and TenGigabitEthernet 0/7 pair, configure 4 EtherChannels (1 EC for each ASA). These EtherChannels must all be on the same isolated cluster control VLAN, for example VLAN 101.

Purpose	Connect Interfaces on each of 4 ASAs	To Switch Ports
Inside and outside interfaces	TenGigabitEthernet 0/8 and TenGigabitEthernet 0/9	8 ports total Configure a single EtherChannel (across all ASAs). On the switch, configure these VLANs and networks now; for example, a trunk including VLAN 200 for the inside and VLAN 201 for the outside.
Management interface	Management 0/0	4 ports total Place all interfaces on the same isolated management VLAN, for example VLAN 100.

Configure the Cluster Interface Mode on Each Unit

You can only configure one type of interface for clustering: Spanned EtherChannels or Individual interfaces; you cannot mix interface types in a cluster.

Before you begin

- You must set the mode separately on each ASA that you want to add to the cluster.
- You can always configure the management-only interface as an Individual interface (recommended), even in Spanned EtherChannel mode. The management interface can be an Individual interface even in transparent firewall mode.
- In Spanned EtherChannel mode, if you configure the management interface as an Individual interface, you cannot enable dynamic routing for the management interface. You must use a static route.
- In multiple context mode, you must choose one interface type for all contexts. For example, if you have a mix of transparent and routed mode contexts, you must use Spanned EtherChannel mode for all contexts because that is the only interface type allowed for transparent mode.

Procedure

Step 1 Show any incompatible configuration so that you can force the interface mode and fix your configuration later; the mode is not changed with this command:

cluster interface-mode {**individual** | **spanned**} **check-details**

Example:

```
ciscoasa(config)# cluster interface-mode spanned check-details
```

Step 2 Set the interface mode for clustering:

cluster interface-mode {**individual** | **spanned**} **force**

Example:

```
ciscoasa(config)# cluster interface-mode spanned force
```

There is no default setting; you must explicitly choose the mode. If you have not set the mode, you cannot enable clustering.

The **force** option changes the mode without checking your configuration for incompatible settings. You need to manually fix any configuration issues after you change the mode. Because any interface configuration can only be fixed after you set the mode, we recommend using the **force** option so that you can at least start from the existing configuration. You can re-run the **check-details** option after you set the mode for more guidance.

Without the **force** option, if there is any incompatible configuration, you are prompted to clear your configuration and reload, thus requiring you to connect to the console port to reconfigure your management access. If your configuration is compatible (rare), the mode is changed and the configuration is preserved. If you do not want to clear your configuration, you can exit the command by typing **n**.

To remove the interface mode, enter the **no cluster interface-mode** command.

Configure Interfaces on the Control Unit

You must modify any interface that is currently configured with an IP address to be cluster-ready before you enable clustering. For other interfaces, you can configure them before or after you enable clustering; we recommend pre-configuring all of your interfaces so that the complete configuration is synced to new cluster members.

This section describes how to configure interfaces to be compatible with clustering. You can configure data interfaces as either Spanned EtherChannels or as Individual interfaces. Each method uses a different load-balancing mechanism. You cannot configure both types in the same configuration, with the exception of the management interface, which can be an Individual interface even in Spanned EtherChannel mode.

Configure Individual Interfaces (Recommended for the Management Interface)

Individual interfaces are normal routed interfaces, each with their own IP address taken from a pool of IP addresses. The Main cluster IP address is a fixed address for the cluster that always belongs to the current primary unit.

In Spanned EtherChannel mode, we recommend configuring the management interface as an Individual interface. Individual management interfaces let you connect directly to each unit if necessary, while a Spanned EtherChannel interface only allows connection to the current primary unit.

Before you begin

- Except for the management-only interface, you must be in Individual interface mode.
- For multiple context mode, perform this procedure in each context. If you are not already in the context configuration mode, enter the **changeto context** *name* command.
- Individual interfaces require you to configure load balancing on neighbor devices. External load balancing is not required for the management interface.

- (Optional) Configure the interface as a device-local EtherChannel interface, a redundant interface, and/or configure subinterfaces.
  - For an EtherChannel, this EtherChannel is local to the unit, and is not a Spanned EtherChannel.
  - Management-only interfaces cannot be redundant interfaces.

Procedure

Step 1 Configure a pool of Local IP addresses (IPv4 and/or IPv6), one of which will be assigned to each cluster unit for the interface:

(IPv4)

ip local pool *poolname first-address* **—** *last-address* [**mask** *mask*]

(IPv6)

ipv6 local pool *poolname ipv6-address*/*prefix-length number_of_addresses*

Example:

```
ciscoasa(config)# ip local pool ins 192.168.1.2-192.168.1.9
ciscoasa(config-if)# ipv6 local pool insipv6 2001:DB8::1002/32 8
```

Include at least as many addresses as there are units in the cluster. If you plan to expand the cluster, include additional addresses. The Main cluster IP address that belongs to the current primary unit is *not* a part of this pool; be sure to reserve an IP address on the same network for the Main cluster IP address.

You cannot determine the exact Local address assigned to each unit in advance; to see the address used on each unit, enter the **show ip**[**v6**] **local pool** *poolname* command. Each cluster member is assigned a member ID when it joins the cluster. The ID determines the Local IP used from the pool.

Step 2 Enter interface configuration mode:

interface *interface_id*

Example:

```
ciscoasa(config)# interface tengigabitethernet 0/8
```

Step 3 (Management interface only) Set an interface to management-only mode so that it does not pass through traffic:

management-only

By default, Management type interfaces are configured as management-only. In transparent mode, this command is always enabled for a Management type interface.

This setting is required if the cluster interface mode is Spanned.

Step 4 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif inside
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value.

Step 5 Set the Main cluster IP address and identify the cluster pool:

(IPv4)

ip address *ip_address* [*mask*] **cluster-pool** *poolname*

(IPv6)

ipv6 address *ipv6-address*/*prefix-length* **cluster-pool** *poolname*

Example:

```
ciscoasa(config-if)# ip address 192.168.1.1 255.255.255.0 cluster-pool ins
ciscoasa(config-if)# ipv6 address 2001:DB8::1002/32 cluster-pool insipv6
```

This IP address must be on the same network as the cluster pool addresses, but not be part of the pool. You can configure an IPv4 and/or an IPv6 address.

DHCP, PPPoE, and IPv6 autoconfiguration are not supported; you must manually configure the IP addresses.

Step 6 Set the security level, where *number* is an integer between 0 (lowest) and 100 (highest):

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 100
```

Step 7 Enable the interface:

no shutdown

Examples

The following example configures the Management 0/0 and Management 0/1 interfaces as a device-local EtherChannel, and then configures the EtherChannel as an Individual interface:

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8:45:1002/64 8
interface management 0/0

channel-group 1 mode active
no shutdown

interface management 0/1

channel-group 1 mode active
no shutdown

interface port-channel 1
```

```
nameif management
ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
ipv6 address 2001:DB8:45:1001/64 cluster-pool mgmtipv6
security-level 100
management-only
```

Configure Spanned EtherChannels

A Spanned EtherChannel spans all ASAs in the cluster, and provides load balancing as part of the EtherChannel operation.

Before you begin

- You must be in Spanned EtherChannel interface mode.
- For multiple context mode, start this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.
- For transparent mode, configure the bridge group. See Configure the Bridge Virtual Interface (BVI), on page 458.
- *Do not* specify the maximum and minimum links in the EtherChannel—We recommend that you do not specify the maximum and minimum links in the EtherChannel (The **lacp max-bundle** and **port-channel min-bundle** commands) on either the ASA or the switch. If you need to use them, note the following:
  - The maximum links set on the ASA is the total number of active ports for the whole cluster. Be sure the maximum links value configured on the switch is not larger than the ASA value.
  - The minimum links set on the ASA is the minimum active ports to bring up a port-channel interface *per unit*. On the switch, the minimum links is the minimum links across the cluster, so this value will not match the ASA value.
- *Do not* change the load-balancing algorithm from the default (see the **port-channel load-balance** command). On the switch, we recommend that you use one of the following algorithms: **source-dest-ip** or **source-dest-ip-port** (see the Cisco Nexus OS and Cisco IOS **port-channel load-balance** command). Do not use a **vlan** keyword in the load-balance algorithm because it can cause unevenly distributed traffic to the ASAs in a cluster.
- The **lacp port-priority** and **lacp system-priority** commands are not used for a Spanned EtherChannel.
- When using Spanned EtherChannels, the port-channel interface will not come up until clustering is fully enabled. This requirement prevents traffic from being forwarded to a unit that is not an active unit in the cluster.

Procedure

Step 1 Specify the interface you want to add to the channel group:

interface *physical_interface*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

The *physical_interface* ID includes the type, slot, and port number as type slot/port. This first interface in the channel group determines the type and speed for all other interfaces in the group.

Step 2 Assign this interface to an EtherChannel:

channel-group *channel_id* **mode active** [**vss-id** {**1** | **2**}]

Example:

```
ciscoasa(config-if)# channel-group 1 mode active
```

The *channel_id* is between 1 and 48. If the port-channel interface for this channel ID does not yet exist in the configuration, one will be added automatically:

interface port-channel *channel_id*

Only **active** mode is supported for Spanned EtherChannels.

If you are connecting the ASA to two switches in a VSS or vPC, then configure the **vss-id** keyword to identify to which switch this interface is connected (1 or 2). You must also use the **port-channel span-cluster vss-load-balance** command for the port-channel interface in Step 6.

Step 3 Enable the interface:

no shutdown

Step 4 (Optional) Add additional interfaces to the EtherChannel by repeating the process.

Example:

```
ciscoasa(config)# interface gigabitethernet 0/1
ciscoasa(config-if)# channel-group 1 mode active
ciscoasa(config-if)# no shutdown
```

Multiple interfaces in the EtherChannel per unit are useful for connecting to switches in a VSS or vPC. Keep in mind that by default, a spanned EtherChannel can have only 8 active interfaces out of 16 maximum across all members in the cluster; the remaining 8 interfaces are on standby in case of link failure. To use more than 8 active interfaces (but no standby interfaces), disable dynamic port priority using the **clacp static-port-priority** command. When you disable dynamic port priority, you can use up to 32 active links across the cluster. For example, for a cluster of 16 ASAs, you can use a maximum of 2 interfaces on each ASA, for a total of 32 interfaces in the spanned EtherChannel.

Step 5 Specify the port-channel interface:

interface port-channel *channel_id*

Example:

```
ciscoasa(config)# interface port-channel 1
```

This interface was created automatically when you added an interface to the channel group.

Step 6 Set this EtherChannel as a Spanned EtherChannel:

port-channel span-cluster [**vss-load-balance**]

Example:

```
ciscoasa(config-if)# port-channel span-cluster
```

If you are connecting the ASA to two switches in a VSS or vPC, then you should enable VSS load balancing by using the **vss-load-balance** keyword. This feature ensures that the physical link connections between the ASAs to the VSS (or vPC) pair are balanced. You must configure the **vss-id** keyword in the **channel-group** command for each member interface before enabling load balancing (see Step 2).

Step 7 (Optional) You can set the Ethernet properties for the port-channel interface to override the properties set on the Individual interfaces.

This method provides a shortcut to set these parameters because these parameters must match for all interfaces in the channel group.

Step 8 (Optional) If you are creating VLAN subinterfaces on this EtherChannel, do so now.

Example:

```
ciscoasa(config)# interface port-channel 1.10
ciscoasa(config-if)# vlan 10
```

The rest of this procedure applies to the subinterfaces.

Step 9 (Multiple Context Mode) Allocate the interface to a context. Then enter:

```
changeto context name
interface port-channel channel_id
```

Example:

```
ciscoasa(config)# context admin
ciscoasa(config)# allocate-interface port-channel1
ciscoasa(config)# changeto context admin
ciscoasa(config-if)# interface port-channel 1
```

For multiple context mode, the rest of the interface configuration occurs within each context.

Step 10 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif inside
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value.

Step 11 Perform one of the following, depending on the firewall mode.

- Routed Mode—Set the IPv4 and/or IPv6 address:

  (IPv4)

ip address *ip_address* [*mask*]

(IPv6)

ipv6 address *ipv6-prefix*/*prefix-length*

Example:

```
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
ciscoasa(config-if)# ipv6 address 2001:DB8::1001/32
```

DHCP, PPPoE, and IPv6 autoconfig are not supported.

- Transparent Mode—Assign the interface to a bridge group:

  bridge-group *number*

  Example:

  ```
  ciscoasa(config-if)# bridge-group 1
  ```

  Where *number* is an integer between 1 and 100. You can assign up to 4 interfaces to a bridge group. You cannot assign the same interface to more than one bridge group. Note that the BVI configuration includes the IP address.

Step 12 Set the security level:

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 50
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 13 Configure a global MAC address for a Spanned EtherChannel to avoid potential network connectivity problems:

mac-address *mac_address*

Example:

```
ciscoasa(config-if)# mac-address 000C.F142.4CDE
```

With a manually-configured MAC address, the MAC address stays with the current control unit. If you do not configure a MAC address, then if the control unit changes, the new control unit uses a new MAC address for the interface, which can cause a temporary network outage.

In multiple context mode, if you share an interface between contexts, you should instead enable auto-generation of MAC addresses so you do not need to set the MAC address manually. Note that you must manually configure the MAC address using this command for *non-shared* interfaces.

The *mac_address* is in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE is entered as 000C.F142.4CDE.

The first two bytes of a manual MAC address cannot be A2 if you also want to use auto-generated MAC addresses.

Create the Bootstrap Configuration

Each unit in the cluster requires a bootstrap configuration to join the cluster.

Configure the Control Unit Bootstrap Settings

Each unit in the cluster requires a bootstrap configuration to join the cluster. Typically, the first unit you configure to join the cluster will be the control unit. After you enable clustering, after an election period, the cluster elects a control unit. With only one unit in the cluster initially, that unit will become the control unit. Subsequent units that you add to the cluster will be data units.

Before you begin

- Back up your configurations in case you later want to leave the cluster, and need to restore your configuration.
- For multiple context mode, complete these procedures in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.
- Enable jumbo frame reservation for use with the cluster control link, so you can set the cluster control link MTU to the recommended value. Enabling jumbo frames causes the ASA to reload, so you must perform this step before continuing with this procedure.
- You must use the console port to enable or disable clustering. You cannot use Telnet or SSH.
- With the exception of the cluster control link, any interfaces in your configuration must be configured with a cluster IP pool or as a Spanned EtherChannel before you enable clustering, depending on your interface mode. If you have pre-existing interface configuration, you can either clear the interface configuration (**clear configure interface**), or convert your interfaces to cluster interfaces before you enable clustering.
- When you add a unit to a running cluster, you may see temporary, limited packet/connection drops; this is expected behavior.
- Pre-determine the size of the cluster control link. See Size the Cluster Control Link, on page 283.

Procedure

Step 1 Enable the cluster control link interface before you join the cluster.

You will later identify this interface as the cluster control link when you enable clustering.

We recommend that you combine multiple cluster control link interfaces into an EtherChannel if you have enough interfaces. The EtherChannel is local to the ASA, and is not a Spanned EtherChannel.

The cluster control link interface configuration is not replicated from the control unit to data units; however, you must use the same configuration on each unit. Because this configuration is not replicated, you must configure the cluster control link interfaces separately on each unit.

- You cannot use a VLAN subinterface as the cluster control link.
- You cannot use a Management *x/x* interface as the cluster control link, either alone or as an EtherChannel.
- For the ASA 5585-X with an ASA FirePOWER module, Cisco recommends that you use ASA interfaces for the cluster control link, and not interfaces on the ASA FirePOWER module. Module interfaces can drop traffic for up to 30 seconds during a module reload, including reloads that occur during a software upgrade. However, if needed, you can use module interfaces and ASA interfaces in the same cluster control link EtherChannel. When the module interfaces drop, the remaining interfaces in the EtherChannel are still up. The ASA 5585-X Network Module does not run a separate operating system, so it is not affected by this issue.

a) Enter interface configuration mode:

interface *interface_id*

Example:

```
ciscoasa(config)# interface tengigabitethernet 0/6
```

b) (Optional, for an EtherChannel) Assign this physical interface to an EtherChannel:

channel-group *channel_id* **mode on**

Example:

```
ciscoasa(config-if)# channel-group 1 mode on
```

The *channel_id* is between 1 and 48. If the port-channel interface for this channel ID does not yet exist in the configuration, one will be added automatically:

interface port-channel *channel_id*

We recommend using the On mode for cluster control link member interfaces to reduce unnecessary traffic on the cluster control link. The cluster control link does not need the overhead of LACP traffic because it is an isolated, stable network. **Note:** We recommend setting *data* EtherChannels to Active mode.

c) Enable the interface:

no shutdown

You only need to enable the interface; do not configure a name for the interface, or any other parameters.

d) (For an EtherChannel) Repeat for each additional interface you want to add to the EtherChannel:

Example:

```
ciscoasa(config)# interface tengigabitethernet 0/7
ciscoasa(config-if)# channel-group 1 mode on
ciscoasa(config-if)# no shutdown
```

Step 2 Specify the maximum transmission unit for the cluster control link interface to be at least 100 bytes higher than the highest MTU of the data interfaces.

mtu cluster *bytes*

Example:

```
ciscoasa(config)# mtu cluster 9198
```

Set the MTU between 1400 and 9198 bytes. The default MTU is 1500 bytes. We suggest setting the cluster control link MTU to the maximum, which requires you to enable jumbo frame reservation before continuing with this procedure. Jumbo frame reservation requires a reload of the ASA. Because the cluster control link traffic includes data packet forwarding, the cluster control link needs to accommodate the entire size of a data packet plus cluster traffic overhead.

For example, because the maximum MTU is 9198 bytes, then the highest data interface MTU can be 9098, while the cluster control link can be set to 9198.

This command is a global configuration command, but is also part of the bootstrap configuration that is not replicated between units.

Step 3 Name the cluster and enter cluster configuration mode:

cluster group *name*

Example:

```
ciscoasa(config)# cluster group pod1
```

The name must be an ASCII string from 1 to 38 characters. You can only configure one cluster group per unit. All members of the cluster must use the same name.

Step 4 Name this member of the cluster:

local-unit *unit_name*

Use a unique ASCII string from 1 to 38 characters. Each unit must have a unique name. A unit with a duplicated name will be not be allowed in the cluster.

Example:

```
ciscoasa(cfg-cluster)# local-unit unit1
```

Step 5 Specify the cluster control link interface, preferably an EtherChannel:

cluster-interface *interface_id* **ip** *ip_address mask*

Example:

```
ciscoasa(cfg-cluster)# cluster-interface port-channel2 ip 192.168.1.1 255.255.255.0
INFO: Non-cluster interface config is cleared on Port-Channel2
```

Subinterfaces and Management interfaces are not allowed.

Specify an IPv4 address for the IP address; IPv6 is not supported for this interface. This interface cannot have a **nameif** configured.

For each unit, specify a different IP address on the same network.

Step 6 Set the priority of this unit for control unit elections:

priority *priority_number*

Example:

```
ciscoasa(cfg-cluster)# priority 1
```

The priority is between 1 and 100, where 1 is the highest priority.

Step 7 (Optional) Set an authentication key for control traffic on the cluster control link:

key *shared_secret*

Example:

```
ciscoasa(cfg-cluster)# key chuntheunavoidable
```

The shared secret is an ASCII string from 1 to 63 characters. The shared secret is used to generate the key. This command does not affect datapath traffic, including connection state update and forwarded packets, which are always sent in the clear.

Step 8 (Optional) Disable dynamic port priority in LACP:

clacp static-port-priority

Some switches do not support dynamic port priority, so this command improves switch compatibility. Moreover, it enables support of more than 8 active spanned EtherChannel members, up to 32 members. Without this command, only 8 active members and 8 standby members are supported. If you enable this command, then you cannot use any standby members; all members are active.

Step 9 (Optional) Manually specify the cLACP system ID and system priority:

clacp system-mac {*mac_address* | **auto**} [**system-priority** *number*]

Example:

```
ciscoasa(cfg-cluster)# clacp system-mac 000a.0000.aaaa
```

When using Spanned EtherChannels, the ASA uses cLACP to negotiate the EtherChannel with the neighbor switch. ASAs in a cluster collaborate in cLACP negotiation so that they appear as a single (virtual) device to the switch. One parameter in cLACP negotiation is a system ID, which is in the format of a MAC address. All ASAs in the cluster use the same system ID: auto-generated by the control unit (the default) and replicated to all secondaries; or manually specified in this command in the form *H.H.H*, where H is a 16-bit hexadecimal digit. (For example, the MAC address 00-0A-00-00-AA-AA is entered as 000A.0000.AAAA.) You might want to manually configure the MAC address for troubleshooting purposes, for example, so that you can use an easily identified MAC address. Typically, you would use the auto-generated MAC address.

The system priority, between 1 and 65535, is used to decide which unit is in charge of making a bundling decision. By default, the ASA uses priority 1, which is the highest priority. The priority needs to be higher than the priority on the switch.

This command is not part of the bootstrap configuration, and is replicated from the control unit to the data units. However, you cannot change this value after you enable clustering.

Step 10 Enable clustering:

enable [**noconfirm**]

Example:

```
ciscoasa(cfg-cluster)# enable
```

```
INFO: Clustering is not compatible with following commands:
policy-map global_policy
 class inspection_default
 inspect skinny
policy-map global_policy
 class inspection_default
  inspect sip
 Would you like to remove these commands? [Y]es/[N]o:Y

INFO: Removing incompatible commands from running configuration...
Cryptochecksum (changed): f16b7fc2 a742727e e40bc0b0 cd169999
INFO: Done
```

When you enter the **enable** command, the ASA scans the running configuration for incompatible commands for features that are not supported with clustering, including commands that may be present in the default configuration. You are prompted to delete the incompatible commands. If you respond **No**, then clustering is not enabled. Use the **noconfirm** keyword to bypass the confirmation and delete incompatible commands automatically.

For the first unit enabled, a control unit election occurs. Because the first unit should be the only member of the cluster so far, it will become the control unit. Do not perform any configuration changes during this period.

To disable clustering, enter the **no enable** command.

Note If you disable clustering, all data interfaces are shut down, and only the management-only interface is active.

Examples

The following example configures a management interface, configures a device-local EtherChannel for the cluster control link, and then enables clustering for the ASA called "unit1," which will become the control unit because it is added to the cluster first:

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8::1002/32 8
interface management 0/0
   nameif management
   ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
   ipv6 address 2001:DB8::1001/32 cluster-pool mgmtipv6
   security-level 100
   management-only
   no shutdown

interface tengigabitethernet 0/6
   channel-group 1 mode on
   no shutdown

interface tengigabitethernet 0/7
   channel-group 1 mode on
   no shutdown

cluster group pod1
   local-unit unit1
   cluster-interface port-channel1 ip 192.168.1.1 255.255.255.0
   priority 1
   key chuntheunavoidable
```

```
enable noconfirm
```

Configure Data Unit Bootstrap Settings

Perform the following procedure to configure the data units.

Before you begin

- You must use the console port to enable or disable clustering. You cannot use Telnet or SSH.
- Back up your configurations in case you later want to leave the cluster, and need to restore your configuration.
- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.
- Enable jumbo frame reservation for use with the cluster control link, so you can set the cluster control link MTU to the recommended value. Enabling jumbo frames causes the ASA to reload, so you must perform this step before continuing with this procedure.
- If you have any interfaces in your configuration that have not been configured for clustering (for example, the default configuration Management 0/0 interface), you can join the cluster as a data unit (with no possibility of becoming the control unit in a current election).
- When you add a unit to a running cluster, you may see temporary, limited packet/connection drops; this is expected behavior.

Procedure

Step 1 Configure the same cluster control link interface as you configured for the control unit.

Example:

```
ciscoasa(config)# interface tengigabitethernet 0/6
ciscoasa(config-if)# channel-group 1 mode on
ciscoasa(config-if)# no shutdown
ciscoasa(config)# interface tengigabitethernet 0/7
ciscoasa(config-if)# channel-group 1 mode on
ciscoasa(config-if)# no shutdown
```

Step 2 Specify the same MTU that you configured for the control unit:

Example:

```
ciscoasa(config)# mtu cluster 9198
```

Step 3 Identify the same cluster name that you configured for the control unit:

Example:

```
ciscoasa(config)# cluster group pod1
```

Step 4 Name this member of the cluster with a unique string:

local-unit *unit_name*

Example:

```
ciscoasa(cfg-cluster)# local-unit unit2
```

Specify an ASCII string from 1 to 38 characters.

Each unit must have a unique name. A unit with a duplicated name will be not be allowed in the cluster.

Step 5 Specify the same cluster control link interface that you configured for the control unit, but specify a different IP address on the same network for each unit:

cluster-interface *interface_id* **ip** *ip_address mask*

Example:

```
ciscoasa(cfg-cluster)# cluster-interface port-channel2 ip 192.168.1.2 255.255.255.0
INFO: Non-cluster interface config is cleared on Port-Channel2
```

Specify an IPv4 address for the IP address; IPv6 is not supported for this interface. This interface cannot have a **nameif** configured.

Each unit must have a unique name. A unit with a duplicated name will not be allowed in the cluster.

Step 6 Set the priority of this unit for control unit elections, typically to a higher value than the control unit:

priority *priority_number*

Example:

```
ciscoasa(cfg-cluster)# priority 2
```

Set the priority between 1 and 100, where 1 is the highest priority.

Step 7 Set the same authentication key that you set for the control unit:

Example:

```
ciscoasa(cfg-cluster)# key chuntheunavoidable
```

Step 8 Enable clustering:

enable as-slave

You can avoid any configuration incompatibilities (primarily the existence of any interfaces not yet configured for clustering) by using the **enable as-slave** command. This command ensures the data unit joins the cluster with no possibility of becoming the control unit in any current election. Its configuration is overwritten with the one synced from the control unit.

To disable clustering, enter the **no enable** command.

Note If you disable clustering, all data interfaces are shut down, and only the management interface is active.

Examples

The following example includes the configuration for a data unit, unit2:

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown

cluster group pod1

local-unit unit2
cluster-interface port-channel1 ip 192.168.1.2 255.255.255.0
priority 2
key chuntheunavoidable
enable as-slave
```

Customize the Clustering Operation

You can customize clustering health monitoring, TCP connection replication delay, flow mobility and other optimizations.

Perform these procedures on the control unit.

Configure Basic ASA Cluster Parameters

You can customize cluster settings on the control unit.

Before you begin

- For multiple context mode, complete this procedure in the system execution space on the control unit. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Step 2 (Optional) Enable console replication from data units to the control unit:

console-replicate

This feature is disabled by default. The ASA prints out some messages directly to the console for certain critical events. If you enable console replication, data units send the console messages to the control unit so that you only need to monitor one console port for the cluster.

Configure Health Monitoring

This procedure configures unit and interface health monitoring.

You might want to disable health monitoring of non-essential interfaces, for example, the management interface. You can monitor any port-channel ID, redundant ID, or single physical interface ID. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored.

Procedure

Step 1 Enter cluster configuration mode.

cluster group *name*

Example:

```
ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)#
```

Step 2 Customize the cluster unit health check feature.

health-check [**holdtime** *timeout*] [**vss-enabled**]

To determine unit health, the ASA cluster units send keepalive messages on the cluster control link to other units. If a unit does not receive any keepalive messages from a peer unit within the holdtime period, the peer unit is considered unresponsive or dead.

- **holdtime** *timeout*—Determines the amount of time between unit keepalive status messages, between .8 and 45 seconds; The default is 3 seconds.
- **vss-enabled**—Floods the keepalive messages on all EtherChannel interfaces in the cluster control link to ensure that at least one of the switches can receive them. If you configure the cluster control link as an EtherChannel (recommended), and it is connected to a VSS or vPC pair, then you might need to enable the **vss-enabled** option. For some switches, when one unit in the VSS/vPC is shutting down or booting up, EtherChannel member interfaces connected to that switch may appear to be Up to the ASA, but they are not passing traffic on the switch side. The ASA can be erroneously removed from the cluster if you set the ASA holdtime timeout to a low value (such as .8 seconds), and the ASA sends keepalive messages on one of these EtherChannel interfaces.

When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the ASA or the switch, or adding an additional switch to form a VSS or vPC) you should disable the health check feature and also disable interface monitoring for the disabled interfaces (**no health-check monitor-interface**). When the topology change is complete, and the configuration change is synced to all units, you can re-enable the health check feature.

Example:

```
ciscoasa(cfg-cluster)# health-check holdtime 5
```

Step 3 Disable the interface health check on an interface.

no health-check monitor-interface *interface_id*

The interface health check monitors for link failures. If all physical ports for a given logical interface fail on a particular unit, but there are active ports under the same logical interface on other units, then the unit is removed from the cluster. The amount of time before the ASA removes a member from the cluster depends on the type of interface and whether the unit is an established member or is joining the cluster. Health check is enabled by default for all interfaces. You can disable it per interface using the **no** form of this command. You might want to disable health monitoring of non-essential interfaces, for example, the management interface.

- *interface_id*—Disables monitoring of any port-channel ID, redundant ID, or single physical interface ID. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored.

When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the ASA or the switch, or adding an additional switch to form a VSS or vPC) you should disable the health check feature (**no health-check**) and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all units, you can re-enable the health check feature.

Example:

```
ciscoasa(cfg-cluster)# no health-check monitor-interface management0/0
```

Example

The following example configures the health-check holdtime to .3 seconds; enables VSS; disables monitoring on the Ethernet 1/2 interface, which is used for management.

```
ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)# health-check holdtime .3 vss-enabled
ciscoasa(cfg-cluster)# no health-check monitor-interface ethernet1/2
```

Configure Connection Rebalancing

You can configure connection rebalancing. For more information, see Rebalancing New TCP Connections Across the Cluster, on page 353

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Step 2 (Optional) Enable connection rebalancing for TCP traffic:

conn-rebalance [**frequency** *seconds*]

Example:

```
ciscoasa(cfg-cluster)# conn-rebalance frequency 60
```

This command is disabled by default. If enabled, ASAs exchange load information periodically, and offload new connections from more loaded devices to less loaded devices. The frequency, between 1 and 360 seconds, specifies how often the load information is exchanged. The default is 5 seconds.

Do not configure connection rebalancing for inter-site topologies; you do not want connections rebalanced to cluster members at a different site.

Manage Cluster Members

After you deploy the cluster, you can change the configuration and manage cluster members.

Become an Inactive Member

To become an inactive member of the cluster, disable clustering on the unit while leaving the clustering configuration intact.

Note When an ASA becomes inactive (either manually or through a health check failure), all data interfaces are shut down; only the management-only interface can send and receive traffic. To resume traffic flow, re-enable clustering; or you can remove the unit altogether from the cluster. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster (for example, you saved the configuration with clustering disabled), then the management interface is disabled. You must use the console port for any further configuration.

Before you begin

- You must use the console port; you cannot enable or disable clustering from a remote CLI connection.
- For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Example:

```
ciscoasa(config)# cluster group pod1
```

Step 2 Disable clustering:

no enable

If this unit was the control unit, a new control election takes place, and a different member becomes the control unit.

The cluster configuration is maintained, so that you can enable clustering again later.

Deactivate a Unit

To deactivate a member other than the unit you are logged into, perform the following steps.

Note When an ASA becomes inactive, all data interfaces are shut down; only the management-only interface can send and receive traffic. To resume traffic flow, re-enable clustering. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster (for example, if you saved the configuration with clustering disabled), the management interface is disabled. You must use the console port for any further configuration.

Before you begin

For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Remove the unit from the cluster.

cluster remove unit *unit_name*

The bootstrap configuration remains intact, as well as the last configuration synched from the control unit, so that you can later re-add the unit without losing your configuration. If you enter this command on a data unit to remove the control unit, a new control unit is elected.

To view member names, enter **cluster remove unit ?**, or enter the **show cluster info** command.

Example:

```
ciscoasa(config)# cluster remove unit ?

Current active units in the cluster:
asa2

ciscoasa(config)# cluster remove unit asa2
WARNING: Clustering will be disabled on unit asa2. To bring it back
```

```
to the cluster please logon to that unit and re-enable clustering
```

Rejoin the Cluster

If a unit was removed from the cluster, for example for a failed interface or if you manually deactivated a member, you must manually rejoin the cluster.

Before you begin

- You must use the console port to reenable clustering. Other interfaces are shut down.
- For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.
- Make sure the failure is resolved before you try to rejoin the cluster.

Procedure

Step 1 At the console, enter cluster configuration mode:

cluster group *name*

Example:

```
ciscoasa(config)# cluster group pod1
```

Step 2 Enable clustering.

enable

Leave the Cluster

If you want to leave the cluster altogether, you need to remove the entire cluster bootstrap configuration. Because the current configuration on each member is the same (synced from the primary unit), leaving the cluster also means either restoring a pre-clustering configuration from backup, or clearing your configuration and starting over to avoid IP address conflicts.

Before you begin

You must use the console port; when you remove the cluster configuration, all interfaces are shut down, including the management interface and cluster control link. Moreover, you cannot enable or disable clustering from a remote CLI connection.

Procedure

Step 1 For a secondary unit, disable clustering:

cluster group *cluster_name*
no enable

Example:

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)# no enable
```

You cannot make configuration changes while clustering is enabled on a secondary unit.

Step 2 Clear the cluster configuration:

clear configure cluster

The ASA shuts down all interfaces including the management interface and cluster control link.

Step 3 Disable cluster interface mode:

no cluster interface-mode

The mode is not stored in the configuration and must be reset manually.

Step 4 If you have a backup configuration, copy the backup configuration to the running configuration:

copy *backup_cfg* **running-config**

Example:

```
ciscoasa(config)# copy backup_cluster.cfg running-config

Source filename [backup_cluster.cfg]?

Destination filename [running-config]?
ciscoasa(config)#
```

Step 5 Save the configuration to startup:

write memory

Step 6 If you do not have a backup configuration, reconfigure management access. Be sure to change the interface IP addresses, and restore the correct hostname, for example.

Change the Control Unit

Caution The best method to change the control unit is to disable clustering on the control unit, wait for a new control election, and then re-enable clustering. If you must specify the exact unit you want to become the control unit, use the procedure in this section. Note, however, that for centralized features, if you force a control unit change using this procedure, then all connections are dropped, and you have to re-establish the connections on the new control unit.

To change the control unit, perform the following steps.

Before you begin

For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Set a new unit as the control unit:

cluster master unit *unit_name*

Example:

```
ciscoasa(config)# cluster master unit asa2
```

You will need to reconnect to the Main cluster IP address.

To view member names, enter **cluster master unit ?** (to see all names except the current unit), or enter the **show cluster info** command.

Execute a Command Cluster-Wide

To send a command to all members in the cluster, or to a specific member, perform the following steps. Sending a **show** command to all members collects all output and displays it on the console of the current unit. Other commands, such as **capture** and **copy**, can also take advantage of cluster-wide execution.

Procedure

Send a command to all members, or if you specify the unit name, a specific member:

cluster exec [**unit** *unit_name*] *command*

Example:

```
ciscoasa# cluster exec show xlate
```

To view member names, enter **cluster exec unit ?** (to see all names except the current unit), or enter the **show cluster info** command.

Examples

To copy the same capture file from all units in the cluster at the same time to a TFTP server, enter the following command on the control unit:

```
ciscoasa# cluster exec copy /pcap capture: tftp://10.1.1.56/capture1.pcap
```

Multiple PCAP files, one from each unit, are copied to the TFTP server. The destination capture file name is automatically attached with the unit name, such as capture1_asa1.pcap, capture1_asa2.pcap, and so on. In this example, asa1 and asa2 are cluster unit names.

The following sample output for the **cluster exec show port-channel** summary command shows EtherChannel information for each member in the cluster:

```
ciscoasa# cluster exec show port-channel summary
master(LOCAL):***********************************************************
 Number of channel-groups in use: 2
Group  Port-channel  Protocol  Span-cluster  Ports
------+-------------+-----------+-----------------------------------------------
1         Po1              LACP       Yes  Gi0/0(P)
2         Po2              LACP       Yes  Gi0/1(P)
 slave:******************************************************************
 Number of channel-groups in use: 2
Group  Port-channel  Protocol  Span-cluster   Ports
------+-------------+-----------+-----------------------------------------------
1         Po1              LACP       Yes   Gi0/0(P)
2         Po2              LACP       Yes   Gi0/1(P)
```

Monitoring the ASA Cluster

You can monitor and troubleshoot cluster status and connections.

Monitoring Cluster Status

See the following commands for monitoring cluster status:

- **show cluster info** [**health**]

  With no keywords, the **show cluster info** command shows the status of all members of the cluster.

  The **show cluster info health** command shows the current health of interfaces, units, and the cluster overall.

  See the following output for the **show cluster info** command:

```
ciscoasa# show cluster info
Cluster stbu: On
```

```
  This is "C" in state SLAVE
      ID        : 0

            Version   : 9.4(1)
      Serial No.: P3000000025
      CCL IP    : 10.0.0.3
      CCL MAC   : 000b.fcf8.c192
      Last join : 17:08:59 UTC Sep 26 2011
      Last leave: N/A
Other members in the cluster:
  Unit "D" in state SLAVE
      ID        : 1

            Version   : 9.4(1)
      Serial No.: P3000000001
      CCL IP    : 10.0.0.4
      CCL MAC   : 000b.fcf8.c162
      Last join : 19:13:11 UTC Sep 23 2011
      Last leave: N/A
  Unit "A" in state MASTER
      ID        : 2

            Version   : 9.4(1)
      Serial No.: JAB0815R0JY
      CCL IP    : 10.0.0.1
      CCL MAC   : 000f.f775.541e
      Last join : 19:13:20 UTC Sep 23 2011
      Last leave: N/A
  Unit "B" in state SLAVE
      ID        : 3

            Version   : 9.4(1)
      Serial No.: P3000000191
      CCL IP    : 10.0.0.2
      CCL MAC   : 000b.fcf8.c61e
      Last join : 19:13:50 UTC Sep 23 2011
      Last leave: 19:13:36 UTC Sep 23 2011
```

- **show cluster info transport** {**asp** | **cp**}

  Shows transport related statistics for the following:

  - **asp** —Data plane transport statistics.
  - **cp** —Control plane transport statistics.

- **show cluster history**

  Shows the cluster history.

Capturing Packets Cluster-Wide

See the following command for capturing packets in a cluster:

cluster exec capture

To support cluster-wide troubleshooting, you can enable capture of cluster-specific traffic on the control unit using the **cluster exec capture** command, which is then automatically enabled on all of the data units in the cluster.

Monitoring Cluster Resources

See the following command for monitoring cluster resources:

show cluster {**cpu** | **memory** | **resource**} [*options*]

Displays aggregated data for the entire cluster. The *options* available depends on the data type.

Monitoring Cluster Traffic

See the following commands for monitoring cluster traffic:

- **show conn [detail], cluster exec show conn**

  The **show conn** command shows whether a flow is a director, backup, or forwarder flow. Use the **cluster exec show conn** command on any unit to view all connections. This command can show how traffic for a single flow arrives at different ASAs in the cluster. The throughput of the cluster is dependent on the efficiency and configuration of load balancing. This command provides an easy way to view how traffic for a connection is flowing through the cluster, and can help you understand how a load balancer might affect the performance of a flow.

  The following is sample output for the **show conn detail** command:

```
ciscoasa/ASA2/slave# show conn detail
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media,
       M - SMTP data, m - SIP media, n - GUP
       O - outbound data, o - offloaded,
       P - inside back connection,
       Q - Diameter, q - SQL*Net data,
       R - outside acknowledged FIN,
       R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
       V - VPN orphan, W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow
ESP outside: 10.1.227.1/53744 NP Identity Ifc: 10.1.226.1/30604, , flags c, idle 0s,
uptime
1m21s, timeout 30s, bytes 7544, cluster sent/rcvd bytes 0/0, owners (0,255) Traffic
received
at interface outside Locally received: 7544 (93 byte/s) Traffic received at interface
NP
Identity Ifc Locally received: 0 (0 byte/s) UDP outside: 10.1.227.1/500 NP Identity
Ifc:
10.1.226.1/500, flags -c, idle 1m22s, uptime 1m22s, timeout 2m0s, bytes 1580, cluster
sent/rcvd bytes 0/0, cluster sent/rcvd total bytes 0/0, owners (0,255) Traffic received
 at
interface outside Locally received: 864 (10 byte/s) Traffic received at interface NP
Identity
```

```
Ifc Locally received: 716 (8 byte/s)
```

To troubleshoot the connection flow, first see connections on all units by entering the **cluster exec show conn** command on any unit. Look for flows that have the following flags: director (Y), backup (y), and forwarder (z). The following example shows an SSH connection from 172.18.124.187:22 to 192.168.103.131:44727 on all three ASAs; ASA 1 has the z flag showing it is a forwarder for the connection, ASA3 has the Y flag showing it is the director for the connection, and ASA2 has no special flags showing it is the owner. In the outbound direction, the packets for this connection enter the inside interface on ASA2 and exit the outside interface. In the inbound direction, the packets for this connection enter the outside interface on ASA 1 and ASA3, are forwarded over the cluster control link to ASA2, and then exit the inside interface on ASA2.

```
ciscoasa/ASA1/master# cluster exec show conn
ASA1(LOCAL):**********************************************************
18 in use, 22 most used
Cluster stub connections: 0 in use, 5 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags z


ASA2:*****************************************************************
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags UIO


ASA3:*****************************************************************
10 in use, 12 most used
Cluster stub connections: 2 in use, 29 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727, idle 0:00:03, bytes 0,
flags  Y
```

- **show cluster info** [**conn-distribution** | **packet-distribution** | **loadbalance**]

  The **show cluster info conn-distribution** and **show cluster info packet-distribution** commands show traffic distribution across all cluster units. These commands can help you to evaluate and adjust the external load balancer.

  The **show cluster info loadbalance** command shows connection rebalance statistics.

- **show cluster** {**access-list** | **conn** | **traffic** | **user-identity** | **xlate**} [*options*]

  Displays aggregated data for the entire cluster. The *options* available depends on the data type.

  See the following output for the **show cluster access-list** command:

```
ciscoasa# show cluster access-list
hitcnt display order: cluster-wide aggregated result, unit-A, unit-B,  unit-C, unit-D
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval
 300
access-list 101; 122 elements; name hash: 0xe7d586b5
access-list 101 line 1 extended permit tcp 192.168.143.0 255.255.255.0 any eq www
(hitcnt=0, 0, 0, 0, 0) 0x207a2b7d
access-list 101 line 2 extended permit tcp any 192.168.143.0 255.255.255.0 (hitcnt=0,
0, 0, 0, 0) 0xfe4f4947
access-list  101 line 3 extended permit tcp host 192.168.1.183 host 192.168.43.238
(hitcnt=1, 0, 0, 0, 1) 0x7b521307
access-list 101 line 4 extended permit tcp host 192.168.1.116 host 192.168.43.238
(hitcnt=0, 0, 0, 0, 0) 0x5795c069
```

```
access-list 101 line 5 extended permit tcp host 192.168.1.177 host 192.168.43.238
(hitcnt=1, 0, 0, 1, 0) 0x51bde7ee
access list 101 line 6 extended permit tcp host 192.168.1.177 host 192.168.43.13
(hitcnt=0, 0, 0, 0, 0) 0x1e68697c
access-list 101 line 7 extended permit tcp host 192.168.1.177 host 192.168.43.132
(hitcnt=2, 0, 0, 1, 1) 0xc1ce5c49
access-list 101 line 8 extended permit tcp host 192.168.1.177 host 192.168.43.192
(hitcnt=3, 0, 1, 1, 1) 0xb6f59512
access-list 101 line 9 extended permit tcp host 192.168.1.177 host 192.168.43.44
(hitcnt=0, 0, 0, 0, 0) 0xdc104200
access-list 101 line 10 extended permit tcp host 192.168.1.112 host 192.168.43.44
(hitcnt=429, 109, 107, 109, 104)
0xce4f281d
access-list 101 line 11 extended permit tcp host 192.168.1.170 host 192.168.43.238
(hitcnt=3, 1, 0, 0, 2) 0x4143a818
access-list 101 line 12 extended permit tcp host 192.168.1.170 host 192.168.43.169
(hitcnt=2, 0, 1, 0, 1) 0xb18dfea4
access-list 101 line 13 extended permit tcp host 192.168.1.170 host 192.168.43.229
(hitcnt=1, 1, 0, 0, 0) 0x21557d71
access-list 101 line 14 extended permit tcp host 192.168.1.170 host 192.168.43.106
(hitcnt=0, 0, 0, 0, 0) 0x7316e016
access-list 101 line 15 extended permit tcp host 192.168.1.170 host 192.168.43.196
(hitcnt=0, 0, 0, 0, 0) 0x013fd5b8
access-list 101 line 16 extended permit tcp host 192.168.1.170 host 192.168.43.75
(hitcnt=0, 0, 0, 0, 0) 0x2c7dba0d
```

To display the aggregated count of in-use connections for all units, enter:

```
ciscoasa# show cluster conn count
Usage Summary In Cluster:*********************************************
  200 in use (cluster-wide aggregated)
     cl2(LOCAL):***********************************************************
  100 in use, 100 most used

  cl1:******************************************************************
  100 in use, 100 most used
```

- **show asp cluster counter**

  This command is useful for datapath troubleshooting.

Monitoring Cluster Routing

See the following commands for cluster routing:

- **show route cluster**
- **debug route cluster**

  Shows cluster information for routing.

Configuring Logging for Clustering

See the following command for configuring logging for clustering:

logging device-id

Each unit in the cluster generates syslog messages independently. You can use the **logging device-id** command to generate syslog messages with identical or different device IDs to make messages appear to come from the same or different units in the cluster.

Monitoring Cluster Interfaces

See the following commands for monitoring cluster interfaces:

- **show cluster interface-mode**

  Shows the cluster interface mode.

- **show port-channel**

  Includes information about whether a port-channel is spanned.

- **show lacp cluster {system-mac | system-id}**

  Shows the cLACP system ID and priority.

- **debug lacp cluster [all | ccp | misc | protocol]**

  Shows debug messages for cLACP.

Debugging Clustering

See the following commands for debugging clustering:

- **debug cluster [ccp | datapath | fsm | general | hc | license | rpc | transport]**

  Shows debug messages for clustering.

- **show cluster info trace**

  The **show cluster info trace** command shows the debug information for further troubleshooting.

  See the following output for the **show cluster info trace** command:

```
ciscoasa# show cluster info trace
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Send CCP message to all: CCP_MSG_KEEPALIVE from 80-1 at
MASTER
```

Examples for ASA Clustering

These examples include all cluster-related ASA configuration for typical deployments.

Sample ASA and Switch Configuration

The following sample configurations connect the following interfaces between the ASA and the switch:

ASA Interface	Switch Interface
GigabitEthernet 0/2	GigabitEthernet 1/0/15
GigabitEthernet 0/3	GigabitEthernet 1/0/16
GigabitEthernet 0/4	GigabitEthernet 1/0/17
GigabitEthernet 0/5	GigabitEthernet 1/0/18

ASA Configuration

Interface Mode on Each Unit

```
cluster interface-mode spanned force
```

ASA1 Control Unit Bootstrap Configuration

```
interface GigabitEthernet0/0
 channel-group 1 mode on
 no shutdown
!
interface GigabitEthernet0/1
 channel-group 1 mode on
 no shutdown
!
interface Port-channel1
 description Clustering Interface
!
cluster group Moya
 local-unit A
 cluster-interface Port-channel1 ip 10.0.0.1 255.255.255.0
 priority 10
 key emphyri0
 enable noconfirm
```

ASA2 Data Unit Bootstrap Configuration

```
interface GigabitEthernet0/0
 channel-group 1 mode on
 no shutdown
!
interface GigabitEthernet0/1
 channel-group 1 mode on
 no shutdown
!
interface Port-channel1
 description Clustering Interface
!
cluster group Moya
 local-unit B
 cluster-interface Port-channel1 ip 10.0.0.2 255.255.255.0
 priority 11
 key emphyri0
 enable as-slave
```

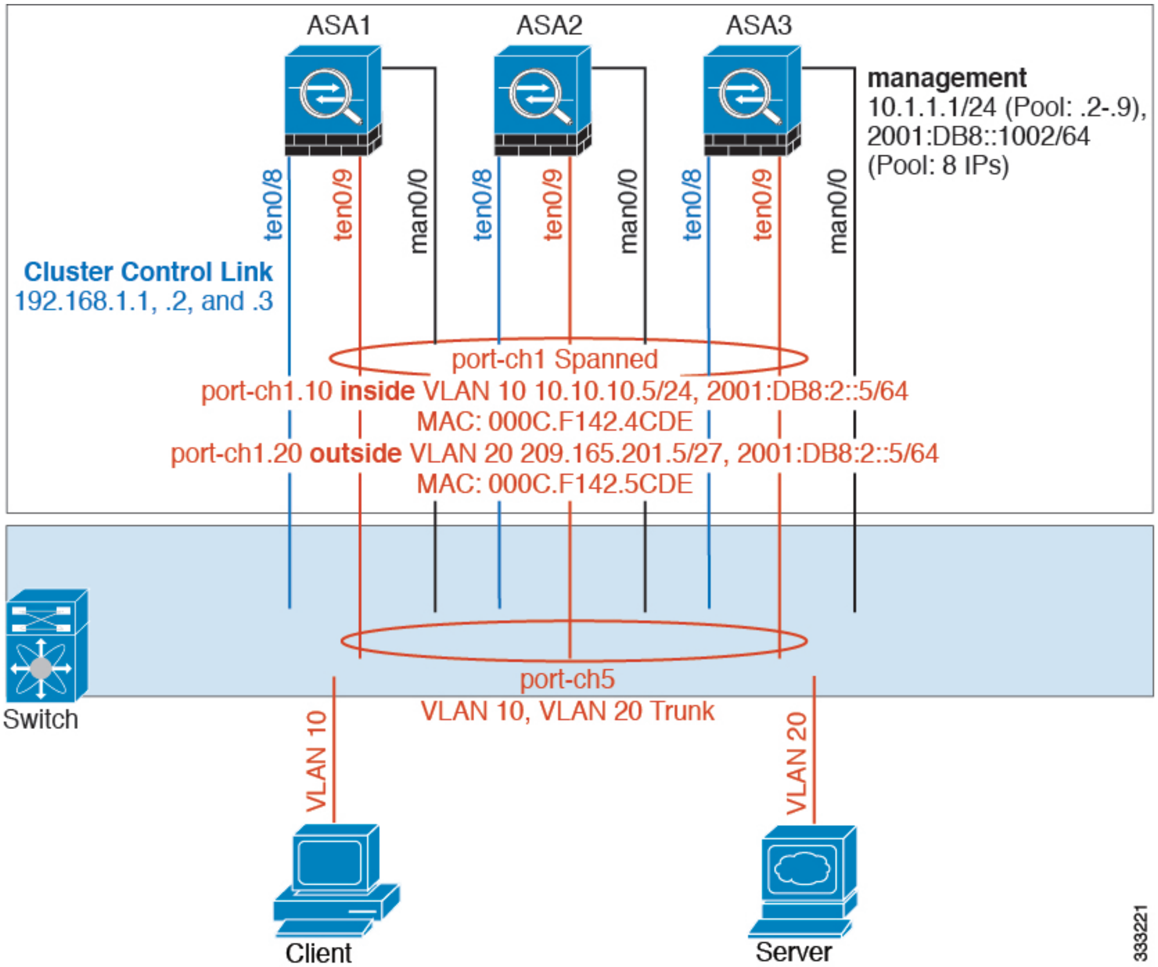
Control Unit Interface Configuration

```
ip local pool mgmt-pool 10.53.195.231-10.53.195.232

interface GigabitEthernet0/2
 channel-group 10 mode active
 no shutdown
!
interface GigabitEthernet0/3
 channel-group 10 mode active
 no shutdown
!
interface GigabitEthernet0/4
 channel-group 11 mode active
 no shutdown
!
interface GigabitEthernet0/5
 channel-group 11 mode active
 no shutdown
!
interface Management0/0
 management-only
 nameif management
 ip address 10.53.195.230 cluster-pool mgmt-pool
 security-level 100
 no shutdown
!
interface Port-channel10
 port-channel span-cluster
 mac-address aaaa.bbbb.cccc
 nameif inside
 security-level 100
 ip address 209.165.200.225 255.255.255.224
!
interface Port-channel11
 port-channel span-cluster
 mac-address aaaa.dddd.cccc
 nameif outside
 security-level 0
 ip address 209.165.201.1 255.255.255.224
```

Cisco IOS Switch Configuration

```
interface GigabitEthernet1/0/15
 switchport access vlan 201
 switchport mode access
 spanning-tree portfast
 channel-group 10 mode active
!
interface GigabitEthernet1/0/16
 switchport access vlan 201
 switchport mode access
 spanning-tree portfast
 channel-group 10 mode active
!
interface GigabitEthernet1/0/17
 switchport access vlan 401
 switchport mode access
 spanning-tree portfast
 channel-group 11 mode active
!
interface GigabitEthernet1/0/18
 switchport access vlan 401
 switchport mode access
 spanning-tree portfast
 channel-group 11 mode active

interface Port-channel10
 switchport access vlan 201
 switchport mode access

interface Port-channel11
 switchport access vlan 401
 switchport mode access
```

Firewall on a Stick



Data traffic from different security domains are associated with different VLANs, for example, VLAN 10 for the inside network and VLAN 20 for the outside network. Each ASA has a single physical port connected to the external switch or router. Trunking is enabled so that all packets on the physical link are 802.1q encapsulated. The ASA is the firewall between VLAN 10 and VLAN 20.

When using Spanned EtherChannels, all data links are grouped into one EtherChannel on the switch side. If an ASA becomes unavailable, the switch will rebalance traffic between the remaining units.

Interface Mode on Each Unit

```
cluster interface-mode spanned force
```

ASA1 Control Unit Bootstrap Configuration

```
interface tengigabitethernet 0/8

no shutdown
description CCL

cluster group cluster1

local-unit asa1
cluster-interface tengigabitethernet0/8 ip 192.168.1.1 255.255.255.0
priority 1
key chuntheunavoidable
enable noconfirm
```

ASA2 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/8

no shutdown
description CCL

cluster group cluster1

local-unit asa2
cluster-interface tengigabitethernet0/8 ip 192.168.1.2 255.255.255.0
priority 2
key chuntheunavoidable
enable as-slave
```

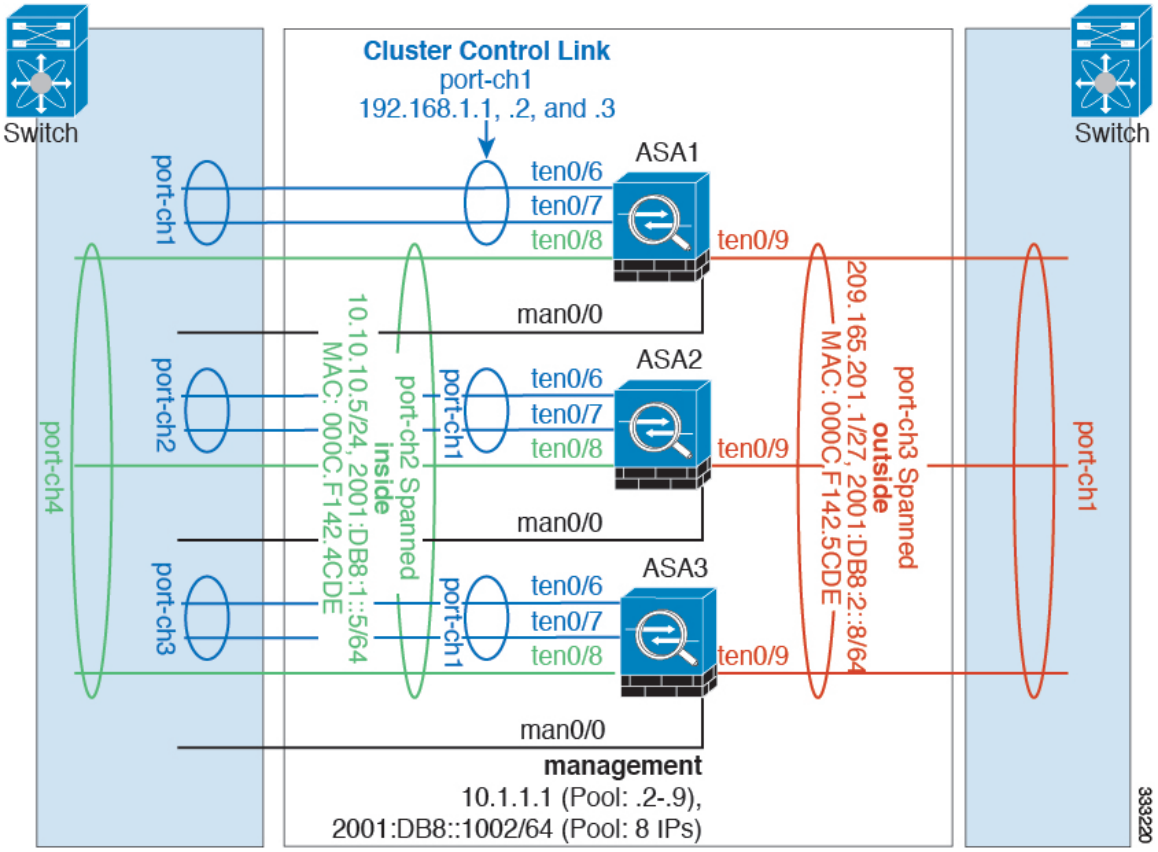
ASA3 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/8

no shutdown
description CCL

cluster group cluster1

local-unit asa3
cluster-interface tengigabitethernet0/8 ip 192.168.1.3 255.255.255.0
priority 3
key chuntheunavoidable
enable as-slave
```

Control Unit Interface Configuration

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8::1002/64 8
interface management 0/0

nameif management
ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
ipv6 address 2001:DB8::1001/32 cluster-pool mgmtipv6
security-level 100
management-only
no shutdown

interface tengigabitethernet 0/9

channel-group 2 mode active
no shutdown
interface port-channel 2
port-channel span-cluster
interface port-channel 2.10
vlan 10
nameif inside
ip address 10.10.10.5 255.255.255.0
ipv6 address 2001:DB8:1::5/64
mac-address 000C.F142.4CDE
interface port-channel 2.20
vlan 20
nameif outside
ip address 209.165.201.1 255.255.255.224
ipv6 address 2001:DB8:2::8/64
mac-address 000C.F142.5CDE
```

Traffic Segregation



You may prefer physical separation of traffic between the inside and outside network.

As shown in the diagram above, there is one Spanned EtherChannel on the left side that connects to the inside switch, and the other on the right side to outside switch. You can also create VLAN subinterfaces on each EtherChannel if desired.

Interface Mode on Each Unit

```
cluster interface-mode spanned force
```

ASA1 Control Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL
```

```
cluster group cluster1

local-unit asa1
cluster-interface port-channel1 ip 192.168.1.1 255.255.255.0
priority 1
key chuntheunavoidable
enable noconfirm
```

ASA2 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL

cluster group cluster1

local-unit asa2
cluster-interface port-channel1 ip 192.168.1.2 255.255.255.0
priority 2
key chuntheunavoidable
enable as-slave
```

ASA3 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL

cluster group cluster1

local-unit asa3
cluster-interface port-channel1 ip 192.168.1.3 255.255.255.0
priority 3
key chuntheunavoidable
enable as-slave
```

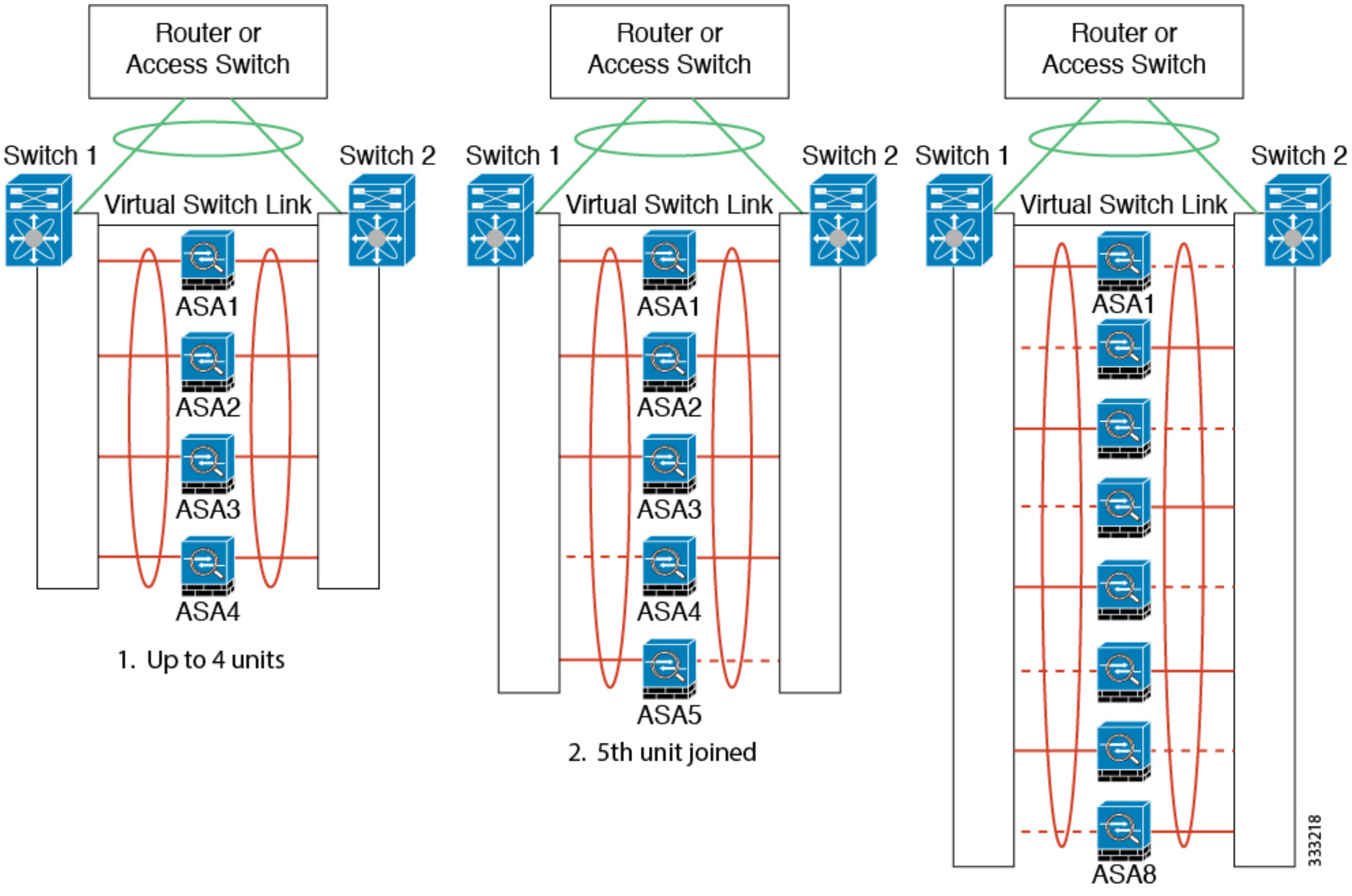
Control Unit Interface Configuration

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8::1002/64 8
```

```
interface management 0/0

nameif management
ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
ipv6 address 2001:DB8::1001/32 cluster-pool mgmtipv6
security-level 100
management-only
no shutdown

interface tengigabitethernet 0/8

channel-group 2 mode active
no shutdown
interface port-channel 2
port-channel span-cluster
nameif inside
ip address 10.10.10.5 255.255.255.0
ipv6 address 2001:DB8:1::5/64
mac-address 000C.F142.4CDE

interface tengigabitethernet 0/9

channel-group 3 mode active
no shutdown
interface port-channel 3
port-channel span-cluster
nameif outside
ip address 209.165.201.1 255.255.255.224
ipv6 address 2001:DB8:2::8/64
mac-address 000C.F142.5CDE
```

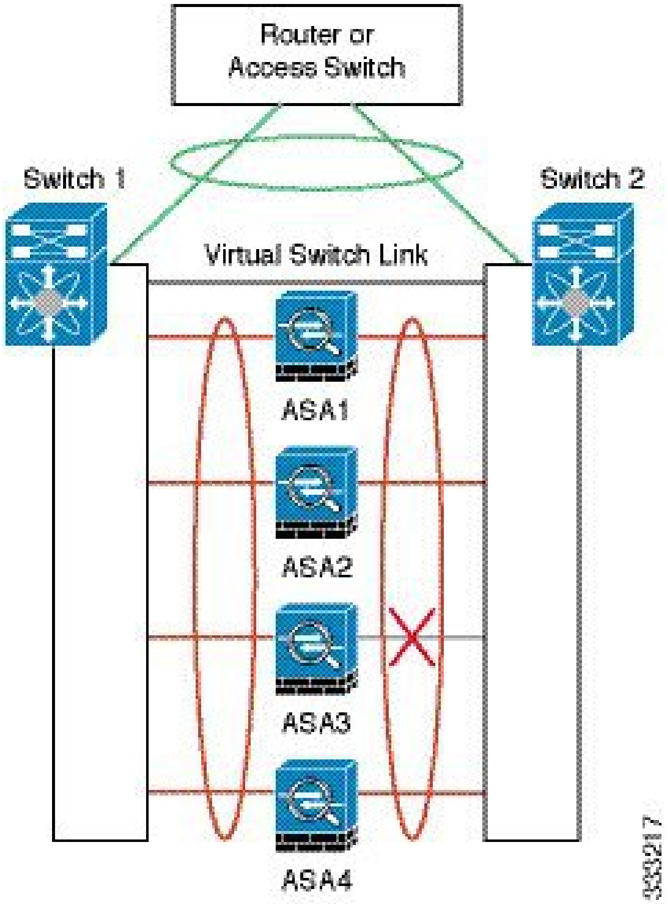
Spanned EtherChannel with Backup Links (Traditional 8 Active/8 Standby)

The maximum number of active ports in a traditional EtherChannel is limited to 8 from the switch side. If you have an 8-ASA cluster, and you allocate 2 ports per unit to the EtherChannel, for a total of 16 ports total, then 8 of them have to be in standby mode. The ASA uses LACP to negotiate which links should be active or standby. If you enable multi-switch EtherChannel using VSS or vPC, you can achieve inter-switch redundancy. On the ASA, all physical ports are ordered first by the slot number then by the port number. In the following figure, the lower ordered port is the “control” port (for example, GigabitEthernet 0/0), and the other one is the “data” port (for example, GigabitEthernet 0/1). You must guarantee symmetry in the hardware connection: all control links must terminate on one switch, and all data links must terminate on another switch if VSS/vPC is used. The following diagram shows what happens when the total number of links grows as more units join the cluster:

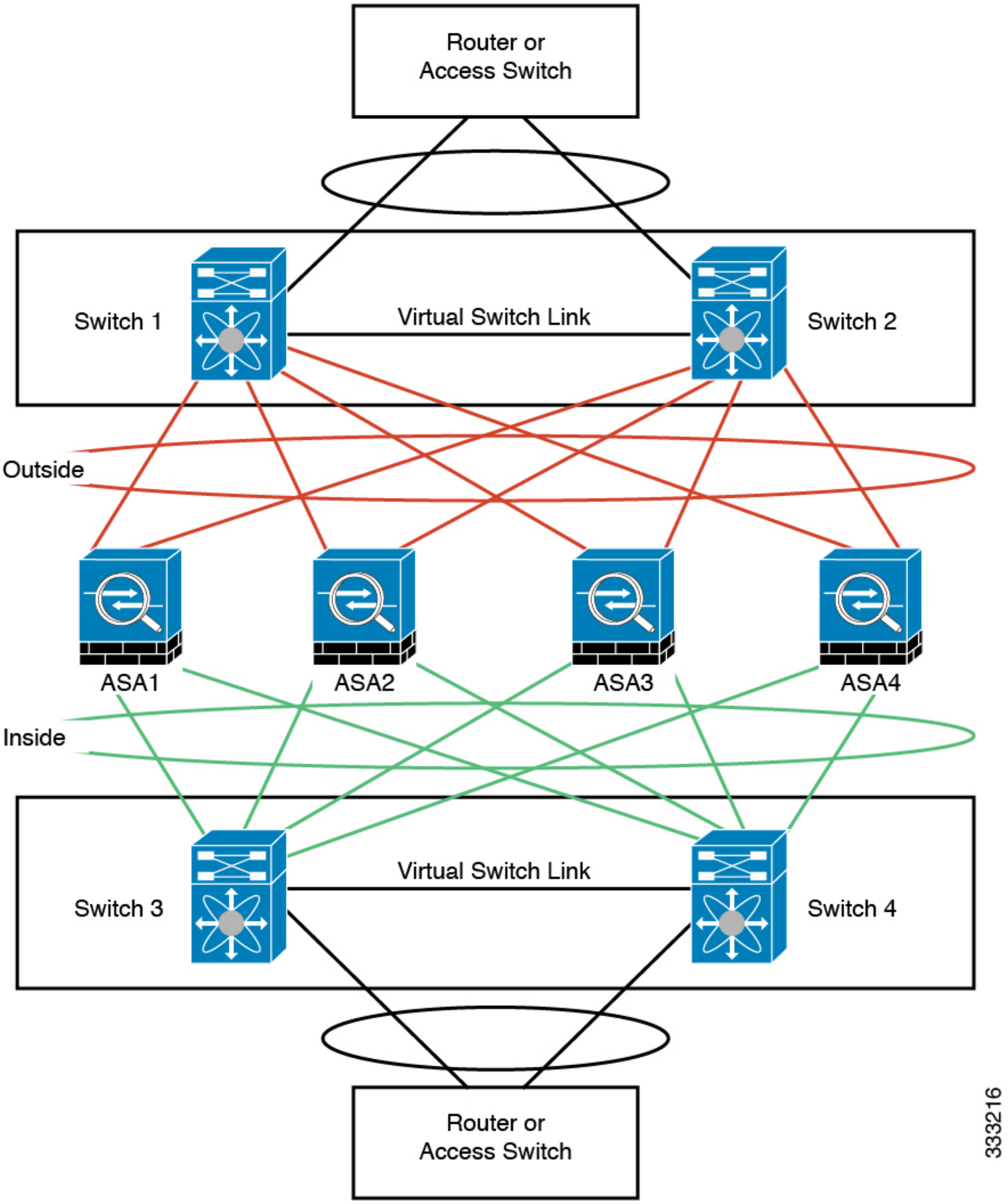


The principle is to first maximize the number of active ports in the channel, and secondly keep the number of active control ports and the number of active data ports in balance. Note that when a 5th unit joins the cluster, traffic is not balanced evenly between all units.

Link or device failure is handled with the same principle. You may end up with a less-than-perfect load balancing situation. The following figure shows a 4-unit cluster with a single link failure on one of the units.



There could be multiple EtherChannels configured in the network. The following diagram shows an EtherChannel on the inside and one on the outside. An ASA is removed from the cluster if both control and data links in one EtherChannel fail. This prevents the ASA from receiving traffic from the outside network when it has already lost connectivity to the inside network.



Interface Mode on Each Unit

```
cluster interface-mode spanned force
```

ASA1 Control Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/8

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/9

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL

cluster group cluster1

local-unit asa1
cluster-interface port-channel1 ip 192.168.1.1 255.255.255.0
priority 1
key chuntheunavoidable
enable noconfirm
```

ASA2 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/8

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/9

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL

cluster group cluster1

local-unit asa2
cluster-interface port-channel1 ip 192.168.1.2 255.255.255.0
priority 2
```

```
key chuntheunavoidable
enable as-slave
```

ASA3 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/8

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/9

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL

cluster group cluster1

local-unit asa3
cluster-interface port-channel1 ip 192.168.1.3 255.255.255.0
priority 3
key chuntheunavoidable
enable as-slave
```

ASA4 Data Unit Bootstrap Configuration

```
interface tengigabitethernet 0/6

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/7

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/8

channel-group 1 mode on
no shutdown

interface tengigabitethernet 0/9

channel-group 1 mode on
no shutdown
interface port-channel 1
description CCL
```

```
cluster group cluster1

local-unit asa4
cluster-interface port-channel1 ip 192.168.1.4 255.255.255.0
priority 4
key chuntheunavoidable
enable as-slave
```

Control Unit Interface Configuration

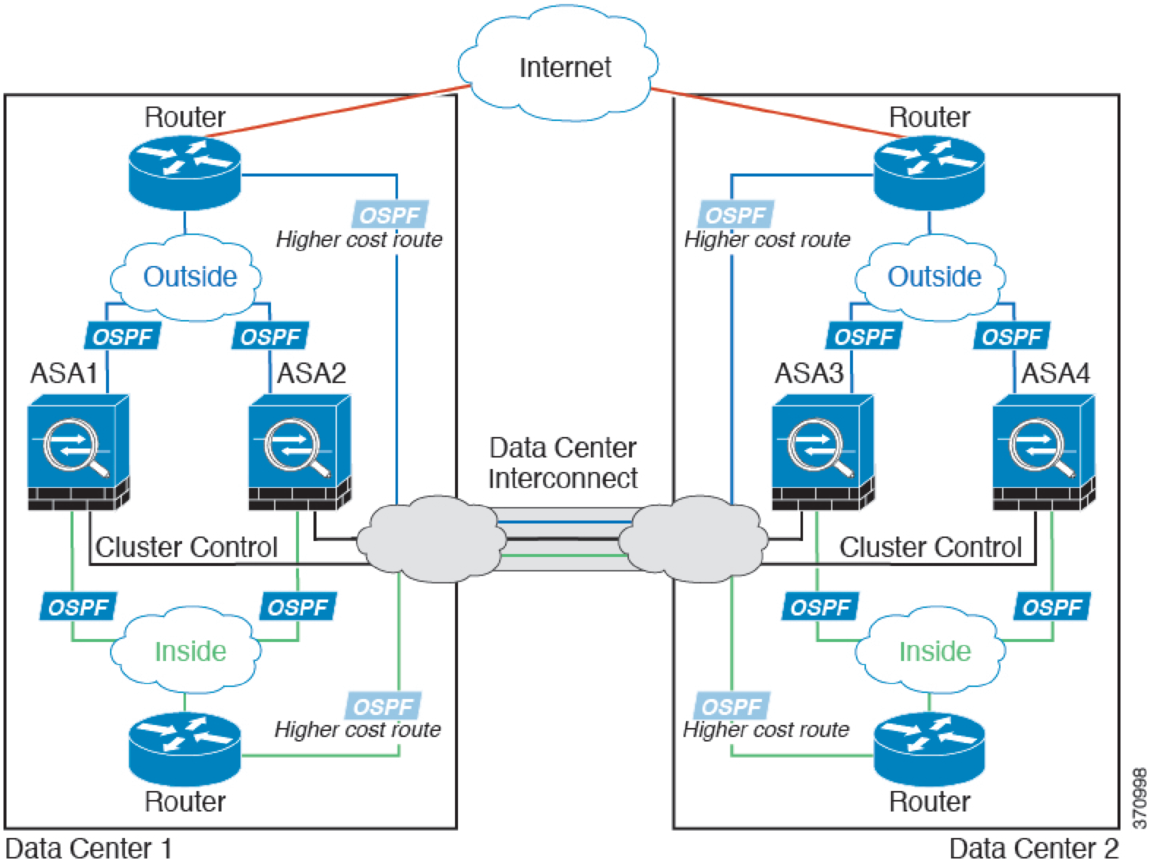
```
ip local pool mgmt 10.1.1.2-10.1.1.9
interface management 0/0

channel-group 2 mode active
no shutdown

interface management 0/1

channel-group 2 mode active
no shutdown
interface port-channel 2
nameif management
ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
security-level 100
management-only

interface tengigabitethernet 1/6

channel-group 3 mode active vss-id 1
no shutdown

interface tengigabitethernet 1/7

channel-group 3 mode active vss-id 2
no shutdown
interface port-channel 3
port-channel span-cluster vss-load-balance
nameif inside
ip address 10.10.10.5 255.255.255.0
mac-address 000C.F142.4CDE

interface tengigabitethernet 1/8

channel-group 4 mode active vss-id 1
no shutdown

interface tengigabitethernet 1/9

channel-group 4 mode active vss-id 2
no shutdown
interface port-channel 4
port-channel span-cluster vss-load-balance
nameif outside
ip address 209.165.201.1 255.255.255.224
mac-address 000C.F142.5CDE
```

Examples for Inter-Site Clustering

The following examples show supported cluster deployments.

Individual Interface Routed Mode North-South Inter-Site Example

The following example shows 2 ASA cluster members at each of 2 data centers placed between inside and outside routers (North-South insertion). The cluster members are connected by the cluster control link over the DCI. The inside and outside routers at each data center use OSPF and PBR or ECMP to load balance the traffic between cluster members. By assigning a higher cost route across the DCI, traffic stays within each data center unless all ASA cluster members at a given site go down. In the event of a failure of all cluster members at one site, traffic goes from each router over the DCI to the ASA cluster members at the other site.



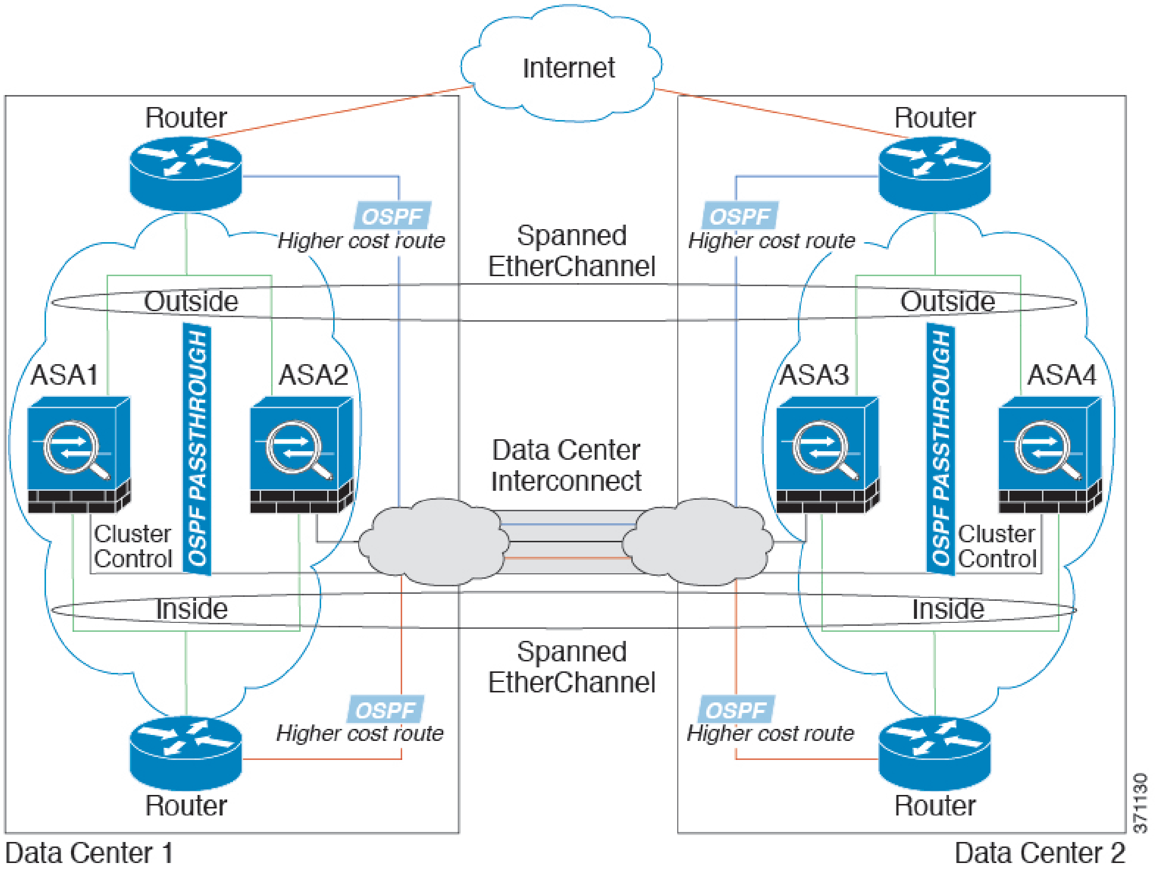
Spanned EtherChannel Transparent Mode North-South Inter-Site Example

The following example shows 2 cluster members at each of 2 data centers placed between inside and outside routers (North-South insertion). The cluster members are connected by the cluster control link over the DCI. The cluster members at each site connect to the local switches using spanned EtherChannels for the inside and outside. Each EtherChannel is spanned across all chassis in the cluster.

The inside and outside routers at each data center use OSPF, which is passed through the transparent ASAs. Unlike MACs, router IPs are unique on all routers. By assigning a higher cost route across the DCI, traffic stays within each data center unless all cluster members at a given site go down. The lower cost route through the ASAs must traverse the same bridge group at each site for the cluster to maintain asymmetric connections. In the event of a failure of all cluster members at one site, traffic goes from each router over the DCI to the cluster members at the other site.

The implementation of the switches at each site can include:

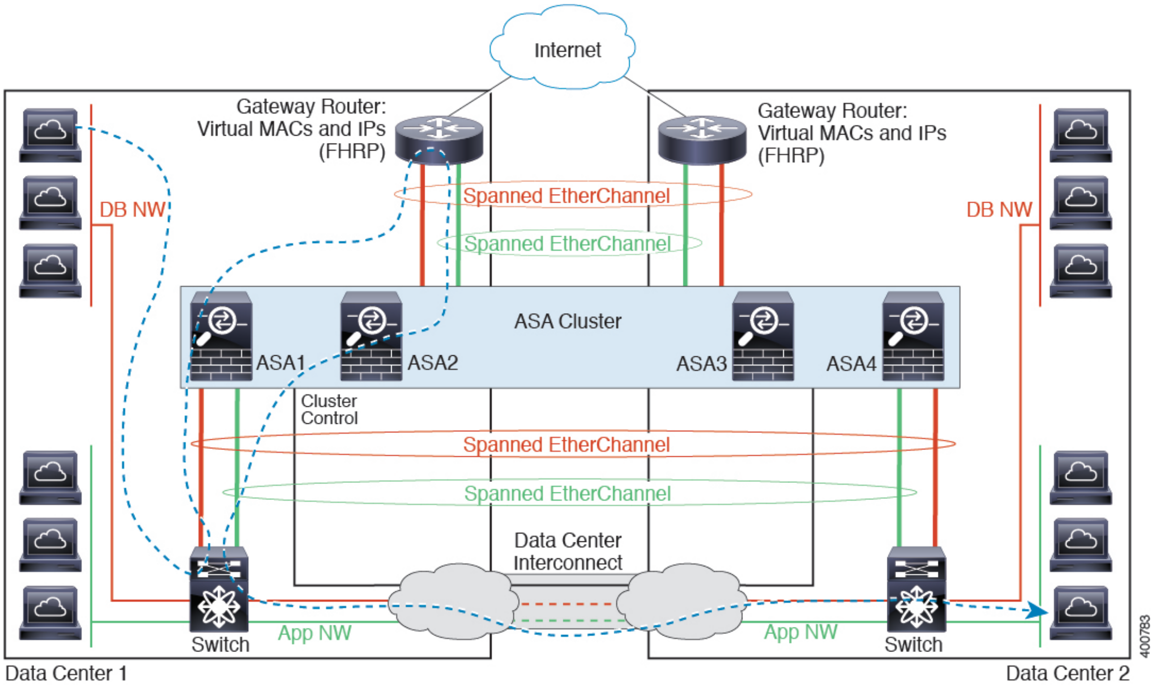
- Inter-site VSS/vPC—In this scenario, you install one switch at Data Center 1, and the other at Data Center 2. One option is for the cluster units at each Data Center to only connect to the local switch, while the VSS/vPC traffic goes across the DCI. In this case, connections are for the most part kept local to each datacenter. You can optionally connect each unit to both switches across the DCI if the DCI can handle the extra traffic. In this case, traffic is distributed across the data centers, so it is essential for the DCI to be very robust.
- Local VSS/vPC at each site—For better switch redundancy, you can install 2 separate VSS/vPC pairs at each site. In this case, although the cluster units still have a spanned EtherChannel with Data Center 1 chassis connected only to both local switches, and Data Center 2 chassis connected to those local switches, the spanned EtherChannel is essentially "split." Each local VSS/vPC sees the spanned EtherChannel as a site-local EtherChannel.



Spanned EtherChannel Transparent Mode East-West Inter-Site Example

The following example shows 2 cluster members at each of 2 data centers placed between the gateway router and two inside networks at each site, the App network and the DB network (East-West insertion). The cluster members are connected by the cluster control link over the DCI. The cluster members at each site connect to the local switches using spanned EtherChannels for both the App and DB networks on the inside and outside. Each EtherChannel is spanned across all chassis in the cluster.

The gateway router at each site uses an FHRP such as HSRP to provide the same destination virtual MAC and IP addresses at each site. A good practice to avoid unintended MAC address flapping is to statically add the gateway routers real MAC addresses to the ASA MAC address table using the **mac-address-table static** *outside_interface mac_address* command. Without these entries, if the gateway at site 1 communicates with the gateway at site 2, that traffic might pass through the ASA and attempt to reach site 2 from the inside interface and cause problems. The data VLANs are extended between the sites using Overlay Transport Virtualization (OTV) (or something similar). You must add filters to prevent traffic from traversing the DCI to the other site when the traffic is destined for the gateway router. If the gateway router at one site becomes unreachable, you must remove the filters so traffic can be sent to the other site's gateway router.



See Spanned EtherChannel Transparent Mode North-South Inter-Site Example, on page 337 for information about vPC/VSS options.

Reference for Clustering

This section includes more information about how clustering operates.

ASA Features and Clustering

Some ASA features are not supported with ASA clustering, and some are only supported on the control unit. Other features might have caveats for proper usage.

Unsupported Features with Clustering

These features cannot be configured with clustering enabled, and the commands will be rejected.

- Unified Communication features that rely on TLS Proxy
- Remote access VPN (SSL VPN and IPsec VPN)
- The following application inspections:

  - CTIQBE
  - GTP
  - H323, H225, and RAS
  - IPsec passthrough
  - MGCP
  - MMP
  - RTSP
  - SCCP (Skinny)
  - WAAS
  - WCCP

- Botnet Traffic Filter
- Auto Update Server
- DHCP client, server, and proxy. DHCP relay is supported.
- VPN load balancing
- Failover
- ASA CX module
- Dead Connection Detection (DCD)
- FIPS mode

Centralized Features for Clustering

The following features are only supported on the control unit, and are not scaled for the cluster. For example, you have a cluster of eight units (5516-X). The Other VPN license allows a maximum of 300 site-to-site IPsec tunnels for one ASA 5516-X. For the entire cluster of eight units, you can only use 300 tunnels; the feature does not scale.

Note Traffic for centralized features is forwarded from member units to the control unit over the cluster control link.

If you use the rebalancing feature, traffic for centralized features may be rebalanced to non-control units before the traffic is classified as a centralized feature; if this occurs, the traffic is then sent back to the control unit.

For centralized features, if the control unit fails, all connections are dropped, and you have to re-establish the connections on the new control unit.

- Site-to-site VPN
- The following application inspections:
  - DCERPC

  - ESMTP
  - IM
  - NetBIOS
  - PPTP
  - RADIUS
  - RSH
  - SNMP
  - SQLNET
  - SUNRPC
  - TFTP
  - XDMCP

- Dynamic routing (Spanned EtherChannel mode only)
- Multicast routing (Individual interface mode only)
- Static route tracking
- IGMP multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- PIM multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- Authentication and Authorization for network access. Accounting is decentralized.
- Filtering Services

Features Applied to Individual Units

These features are applied to each ASA unit, instead of the cluster as a whole or to the control unit.

- QoS—The QoS policy is synced across the cluster as part of configuration replication. However, the policy is enforced on each unit independently. For example, if you configure policing on output, then the conform rate and conform burst values are enforced on traffic exiting a particular ASA. In a cluster with 3 units and with traffic evenly distributed, the conform rate actually becomes 3 times the rate for the cluster.
- Threat detection—Threat detection works on each unit independently; for example, the top statistics is unit-specific. Port scanning detection, for example, does not work because scanning traffic will be load-balanced between all units, and one unit will not see all traffic.
- Resource management—Resource management in multiple context mode is enforced separately on each unit based on local usage.
- ASA Firepower module—There is no configuration sync or state sharing between ASA Firepower modules. You are responsible for maintaining consistent policies on the ASA Firepower modules in the cluster using Firepower Management Center. Do not use different ASA-interface-based zone definitions for devices in the cluster.

- ASA IPS module—There is no configuration sync or state sharing between IPS modules. Some IPS signatures require IPS to keep the state across multiple connections. For example, the port scanning signature is used when the IPS module detects that someone is opening many connections to one server but with different ports. In clustering, those connections will be balanced between multiple ASA devices, each of which has its own IPS module. Because these IPS modules do not share state information, the cluster may not be able to detect port scanning as a result.

AAA for Network Access and Clustering

AAA for network access consists of three components: authentication, authorization, and accounting. Authentication and authorization are implemented as centralized features on the clustering control unit with replication of the data structures to the cluster data units. If a control unit is elected, the new control unit will have all the information it needs to continue uninterrupted operation of the established authenticated users and their associated authorizations. Idle and absolute timeouts for user authentications are preserved when a control unit change occurs.

Accounting is implemented as a distributed feature in a cluster. Accounting is done on a per-flow basis, so the cluster unit owning a flow will send accounting start and stop messages to the AAA server when accounting is configured for a flow.

Connection Settings

Connection limits are enforced cluster-wide (see the **set connection conn-max**, **set connection embryonic-conn-max**, **set connection per-client-embryonic-max**, and **set connection per-client-max** commands). Each unit has an estimate of the cluster-wide counter values based on broadcast messages. Due to efficiency considerations, the configured connection limit across the cluster might not be enforced exactly at the limit number. Each unit may overestimate or underestimate the cluster-wide counter value at any given time. However, the information will get updated over time in a load-balanced cluster.

FTP and Clustering

- If FTP data channel and control channel flows are owned by different cluster members, then the data channel owner will periodically send idle timeout updates to the control channel owner and update the idle timeout value. However, if the control flow owner is reloaded, and the control flow is re-hosted, the parent/child flow relationship will not longer be maintained; the control flow idle timeout will not be updated.
- If you use AAA for FTP access, then the control channel flow is centralized on the control unit.

Identity Firewall and Clustering

Only the control unit retrieves the user-group from the AD and the user-ip mapping from the AD agent. The control unit then populates the user information to data units, and data units can make a match decision for user identity based on the security policy.

Multicast Routing and Clustering

Multicast routing behaves differently depending on the interface mode.

Multicast Routing in Spanned EtherChannel Mode

In Spanned EtherChannel mode, the control unit handles all multicast routing packets and data packets until fast-path forwarding is established. After the connection is established, each data unit can forward multicast data packets.

Multicast Routing in Individual Interface Mode

In Individual interface mode, units do not act independently with multicast. All data and routing packets are processed and forwarded by the control unit, thus avoiding packet replication.

NAT and Clustering

NAT can affect the overall throughput of the cluster. Inbound and outbound NAT packets can be sent to different ASAs in the cluster, because the load balancing algorithm relies on IP addresses and ports, and NAT causes inbound and outbound packets to have different IP addresses and/or ports. When a packet arrives at the ASA that is not the NAT owner, it is forwarded over the cluster control link to the owner, causing large amounts of traffic on the cluster control link. Note that the receiving unit does not create a forwarding flow to the owner, because the NAT owner may not end up creating a connection for the packet depending on the results of security and policy checks.

If you still want to use NAT in clustering, then consider the following guidelines:

- No Proxy ARP—For Individual interfaces, a proxy ARP reply is never sent for mapped addresses. This prevents the adjacent router from maintaining a peer relationship with an ASA that may no longer be in the cluster. The upstream router needs a static route or PBR with Object Tracking for the mapped addresses that points to the Main cluster IP address. This is not an issue for a Spanned EtherChannel, because there is only one IP address associated with the cluster interface.
- No interface PAT on an Individual interface—Interface PAT is not supported for Individual interfaces.
- NAT pool address distribution for dynamic PAT—The control unit evenly pre-distributes addresses across the cluster. If a member receives a connection and they have no addresses assigned, then the connection is forwarded to the control unit for PAT. If a cluster member leaves the cluster (due to failure), a backup member will get the PAT IP address, and if the backup exhausts its normal PAT IP address, it can make use of the new address. Make sure to include at least as many NAT addresses as there are units in the cluster, plus at least one extra address, to ensure that each unit receives an address, and that a failed unit can get a new address if its old address is in use by the member that took over the address. Use the **show nat pool cluster** command to see the address allocations.
- Reusing a PAT pool in multiple rules—To use the same PAT pool in multiple rules, you must be careful about the interface selection in the rules. You must either use specific interfaces in all rules, or "any" in all rules. You cannot mix specific interfaces and "any" across the rules, or the system might not be able to match return traffic to the right node in the cluster. Using unique PAT pools per rule is the most reliable option.
- No round-robin—Round-robin for a PAT pool is not supported with clustering.
- Dynamic NAT xlates managed by the control unit—The control unit maintains and replicates the xlate table to data units. When a data unit receives a connection that requires dynamic NAT, and the xlate is not in the table, it requests the xlate from the control unit. The data unit owns the connection.
- Stale xlates—The xlate idle time on the connection owner does not get updated. Thus, the idle time might exceed the idle timeout. An idle timer value higher than the configured timeout with a refcnt of 0 is an indication of a stale xlate.

- Per-session PAT feature—Although not exclusive to clustering, the per-session PAT feature improves the scalability of PAT and, for clustering, allows each data unit to own PAT connections; by contrast, multi-session PAT connections have to be forwarded to and owned by the control unit. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate, whereas ICMP and all other UDP traffic uses multi-session. You can configure per-session NAT rules to change these defaults for TCP and UDP, but you cannot configure per-session PAT for ICMP. For traffic that benefits from multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT for the associated TCP ports (the UDP ports for those H.323 and SIP are already multi-session by default). For more information about per-session PAT, see the firewall configuration guide.
- No static PAT for the following inspections—
  - FTP
  - PPTP
  - RSH
  - SQLNET
  - TFTP
  - XDMCP
  - SIP
- If you have an extremely large number of NAT rules, over ten thousand, you should enable the transactional commit model using the **asp rule-engine transactional-commit nat** command in the device CLI. Otherwise, the unit might not be able to join the cluster.

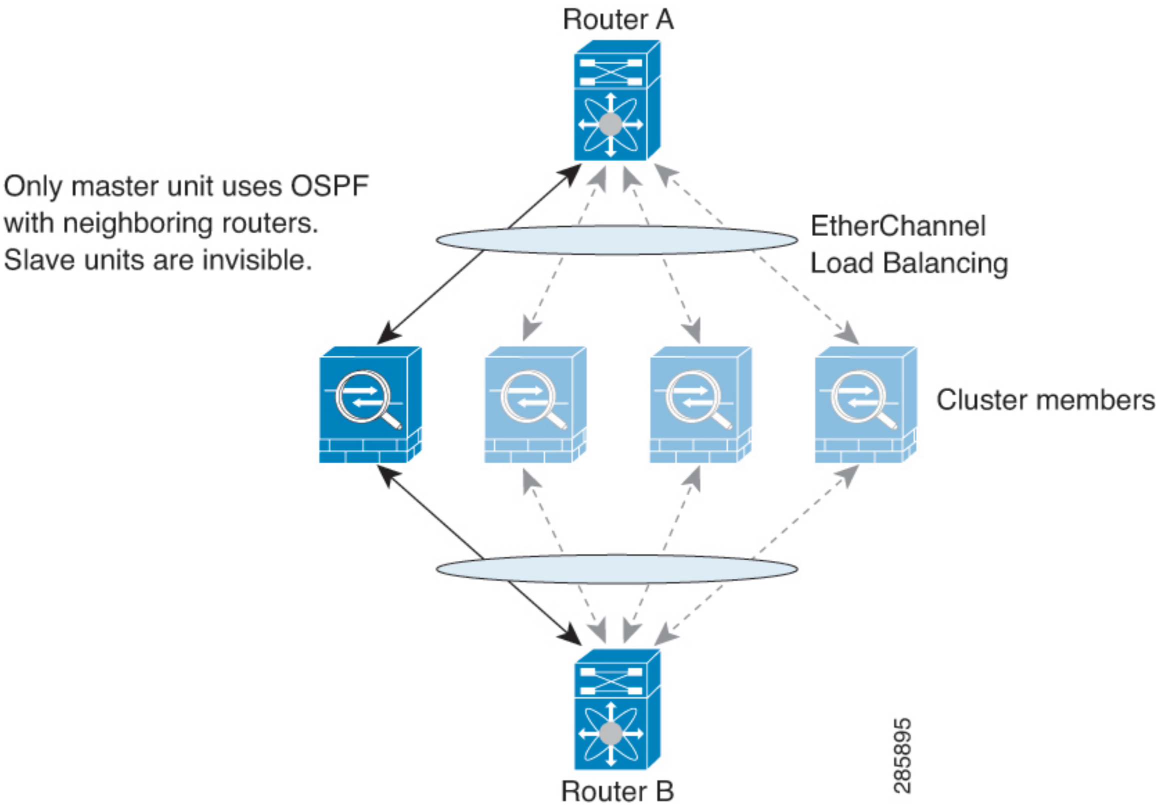
Dynamic Routing and Clustering

This section describes how to use dynamic routing with clustering.

Dynamic Routing in Spanned EtherChannel Mode

In Spanned EtherChannel mode: The routing process only runs on the control unit, and routes are learned through the control unit and replicated to data units. If a routing packet arrives at a data unit, it is redirected to the control unit.

Figure 46: Dynamic Routing in Spanned EtherChannel Mode


Router A
Only master unit uses OSPF
with neighboring routers.
Slave units are invisible.
EtherChannel
Load Balancing
Cluster members
Router B
285895

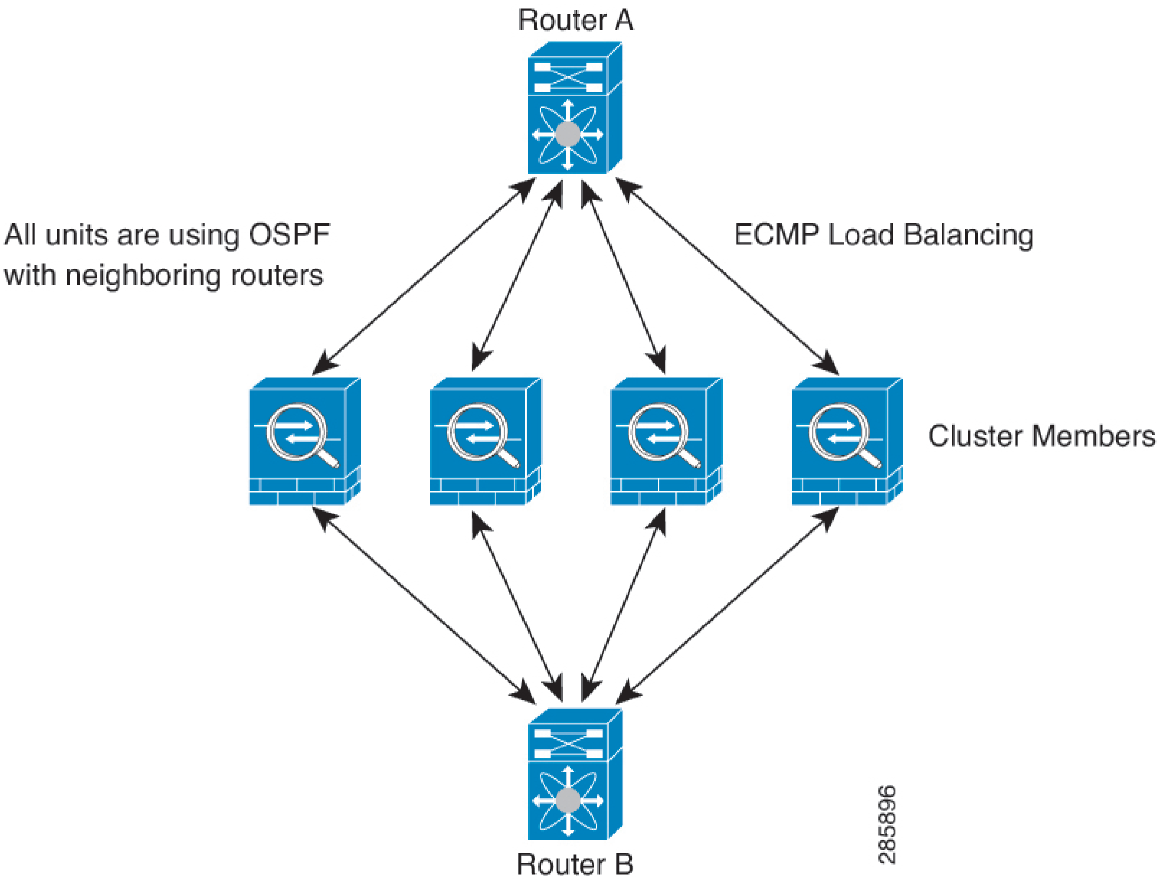
After the data unit learn the routes from the control unit, each unit makes forwarding decisions independently.

The OSPF LSA database is not synchronized from the control unit to data units. If there is a control unit switchover, the neighboring router will detect a restart; the switchover is not transparent. The OSPF process picks an IP address as its router ID. Although not required, you can assign a static router ID to ensure a consistent router ID is used across the cluster. See the OSPF Non-Stop Forwarding feature to address the interruption.

Dynamic Routing in Individual Interface Mode

In Individual interface mode, each unit runs the routing protocol as a standalone router, and routes are learned by each unit independently.

Figure 47: Dynamic Routing in Individual Interface Mode



In the above diagram, Router A learns that there are 4 equal-cost paths to Router B, each through an ASA. ECMP is used to load balance traffic between the 4 paths. Each ASA picks a different router ID when talking to external routers.

You must configure a cluster pool for the router ID so that each unit has a separate router ID.

EIGRP does not form neighbor relationships with cluster peers in individual interface mode.

Note If the cluster has multiple adjacencies to the same router for redundancy purposes, asymmetric routing can lead to unacceptable traffic loss. To avoid asymmetric routing, group all of these ASA interfaces into the same traffic zone. See Configure a Traffic Zone, on page 494.

SIP Inspection and Clustering

A control flow can be created on any unit (due to load balancing); its child data flows must reside on the same unit.

TLS Proxy configuration is not supported.

SNMP and Clustering

An SNMP agent polls each individual ASA by its Local IP address. You cannot poll consolidated data for the cluster.

You should always use the Local address, and not the Main cluster IP address for SNMP polling. If the SNMP agent polls the Main cluster IP address, if a new control unit is elected, the poll to the new control unit will fail.

When using SNMPv3 with clustering, if you add a new cluster unit after the initial cluster formation, then SNMPv3 users are not replicated to the new unit.You must re-add them on the control unit to force the users to replicate to the new unit, or directly on the data unit.

Syslog and NetFlow and Clustering

- Syslog—Each unit in the cluster generates its own syslog messages. You can configure logging so that each unit uses either the same or a different device ID in the syslog message header field. For example, the hostname configuration is replicated and shared by all units in the cluster. If you configure logging to use the hostname as the device ID, syslog messages generated by all units look as if they come from a single unit. If you configure logging to use the local-unit name that is assigned in the cluster bootstrap configuration as the device ID, syslog messages look as if they come from different units.
- NetFlow—Each unit in the cluster generates its own NetFlow stream. The NetFlow collector can only treat each ASA as a separate NetFlow exporter.

Cisco TrustSec and Clustering

Only the control unit learns security group tag (SGT) information. The control unit then populates the SGT to data units, and data units can make a match decision for SGT based on the security policy.

VPN and Clustering

Site-to-site VPN is a centralized feature; only the control unit supports VPN connections. Distributed site-to-site VPN clustering is supported. Search for High Availability options in this pdf for details.

Note Remote access VPN is not supported with clustering.

VPN functionality is limited to the control unit and does not take advantage of the cluster high availability capabilities. If the control unit fails, all existing VPN connections are lost, and VPN users will see a disruption in service. When a new control unit is elected, you must reestablish the VPN connections.

When you connect a VPN tunnel to a Spanned EtherChannel address, connections are automatically forwarded to the control unit. For connections to an Individual interface when using PBR or ECMP, you must always connect to the Main cluster IP address, not a Local address.

VPN-related keys and certificates are replicated to all units.

Performance Scaling Factor

When you combine multiple units into a cluster, you can expect a performance of approximately:

- 70% of the combined throughput

- 60% of maximum connections
- 50% of connections per second

For example, for throughput, the ASA 5585-X with SSP-40 can handle approximately 10 Gbps of real world firewall traffic when running alone. For a cluster of 8 units, the maximum combined throughput will be approximately 70% of 80 Gbps (8 units x 10 Gbps): 56 Gbps.

Control Unit Election

Members of the cluster communicate over the cluster control link to elect a control unit as follows:

1. When you enable clustering for a unit (or when it first starts up with clustering already enabled), it broadcasts an election request every 3 seconds.
2. Any other units with a higher priority respond to the election request; the priority is set between 1 and 100, where 1 is the highest priority.
3. If after 45 seconds, a unit does not receive a response from another unit with a higher priority, then it becomes the control unit.

Note If multiple units tie for the highest priority, the cluster unit name and then the serial number is used to determine the control unit.

4. If a unit later joins the cluster with a higher priority, it does not automatically become the control unit; the existing control unit always remains as the control unit unless it stops responding, at which point a new control unit is elected.
5. In a "split brain" scenario when there are temporarily multiple control units, then the unit with highest priority retains the role while the other units return to data unit roles.

Note You can manually force a unit to become the control unit. For centralized features, if you force a control unit change, then all connections are dropped, and you have to re-establish the connections on the new control unit.

High Availability Within the ASA Cluster

ASA Clustering provides high availability by monitoring unit and interface health and by replicating connection states between units.

Unit Health Monitoring

Each unit periodically sends a broadcast keepalive packet over the cluster control link. If the control unit does not receive any keepalive packets or other packets from a data unit within the configurable timeout period, then the control unit removes the data unit from the cluster. If the data units do not receive packets from the control unit, then a new control unit is elected from the remaining members.

If units cannot reach each other over the cluster control link because of a network failure and not because a unit has actually failed, then the cluster may go into a "split brain" scenario where isolated data units will elect their own control units. For example, if a router fails between two cluster locations, then the original control unit at location 1 will remove the location 2 data units from the cluster. Meanwhile, the units at location 2 will elect their own control unit and form their own cluster. Note that asymmetric traffic may fail in this scenario. After the cluster control link is restored, then the control unit that has the higher priority will keep the control unit's role. See Control Unit Election, on page 348 for more information.

Interface Monitoring

Each unit monitors the link status of all named hardware interfaces in use, and reports status changes to the control unit.

- Spanned EtherChannel—Uses cluster Link Aggregation Control Protocol (cLACP). Each unit monitors the link status and the cLACP protocol messages to determine if the port is still active in the EtherChannel. The status is reported to the control unit.
- Individual interfaces (Routed mode only)—Each unit self-monitors its interfaces and reports interface status to the control unit.

When you enable health monitoring, all physical interfaces (including the main EtherChannel and redundant interface types) are monitored by default; you can optionally disable monitoring per interface. Only named interfaces can be monitored. For example, the named EtherChannel must fail to be considered failed, which means all member ports of an EtherChannel must fail to trigger cluster removal (depending on your minimum port bundling setting).

A unit is removed from the cluster if its monitored interfaces fail. The amount of time before the ASA removes a member from the cluster depends on the type of interface and whether the unit is an established member or is joining the cluster. For EtherChannels (spanned or not), if the interface is down on an established member, then the ASA removes the member after 9 seconds. The ASA does not monitor interfaces for the first 90 seconds that a unit joins the cluster. Interface status changes during this time will not cause the ASA to be removed from the cluster. For non-EtherChannels, the unit is removed after 500 ms, regardless of the member state.

Status After Failure

When a unit in the cluster fails, the connections hosted by that unit are seamlessly transferred to other units; state information for traffic flows is shared over the control unit's cluster control link.

If the control unit fails, then another member of the cluster with the highest priority (lowest number) becomes the control unit.

The ASA automatically tries to rejoin the cluster, depending on the failure event.

Note When the ASA becomes inactive and fails to automatically rejoin the cluster, all data interfaces are shut down; only the management-only interface can send and receive traffic. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster, the management interface is disabled. You must use the console port for any further configuration.

Rejoining the Cluster

After a cluster member is removed from the cluster, how it can rejoin the cluster depends on why it was removed:

- Failed cluster control link—After you resolve the problem with the cluster control link, you must manually rejoin the cluster by re-enabling clustering at the console port by entering **cluster group** *name*, and then **enable**.
- Failed data interface—The ASA automatically tries to rejoin at 5 minutes, then at 10 minutes, and finally at 20 minutes. If the join is not successful after 20 minutes, then the ASA disables clustering. After you resolve the problem with the data interface, you have to manually enable clustering at the console port by entering **cluster group** *name*, and then **enable**.
- Failed ASA FirePOWER module on the ASA 5585-X—The ASA automatically tries to rejoin at 5 minutes.
- Failed ASA FirePOWER software module—After you resolve the problem with the module, you must manually enable clustering at the console port by entering **cluster group** *name*, and then **enable**.
- Failed unit—If the unit was removed from the cluster because of a unit health check failure, then rejoining the cluster depends on the source of the failure. For example, a temporary power failure means the unit will rejoin the cluster when it starts up again as long as the cluster control link is up and clustering is still enabled with the **enable** command. The ASA attempts to rejoin the cluster every 5 seconds.
- Internal error—Internal failures include: application sync timeout; inconsistent application statuses; and so on. After you resolve the problem, you must manually rejoin the cluster by re-enabling clustering at the console port by entering **cluster group** *name*, and then **enable**.

See Configure the Control Unit Bootstrap Settings, on page 302.

Data Path Connection State Replication

Every connection has one owner and at least one backup owner in the cluster. The backup owner does not take over the connection in the event of a failure; instead, it stores TCP/UDP state information, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner is usually also the director.

Some traffic requires state information above the TCP or UDP layer. See the following table for clustering support or lack of support for this kind of traffic.

Table 14: Features Replicated Across the Cluster

Traffic	State Support	Notes
Up time	Yes	Keeps track of the system up time.
ARP Table	Yes	Transparent mode only.
MAC address table	Yes	Transparent mode only.
User Identity	Yes	Includes AAA rules (uauth) and identity firewall.
IPv6 Neighbor database	Yes	—

Traffic	State Support	Notes
Dynamic routing	Yes	—
SNMP Engine ID	**No**	—
Centralized VPN (Site-to-Site)	**No**	VPN sessions will be disconnected if the control unit fails.

How the ASA Cluster Manages Connections

Connections can be load-balanced to multiple members of the cluster. Connection roles determine how connections are handled in both normal operation and in a high availability situation.

Connection Roles

See the following roles defined for each connection:

- Owner—Usually, the unit that initially receives the connection. The owner maintains the TCP state and processes packets. A connection has only one owner. If the original owner fails, then when new units receive packets from the connection, the director chooses a new owner from those units.
- Backup owner—The unit that stores TCP/UDP state information received from the owner, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner does not take over the connection in the event of a failure. If the owner becomes unavailable, then the first unit to receive packets from the connection (based on load balancing) contacts the backup owner for the relevant state information so it can become the new owner.

  As long as the director (see below) is not the same unit as the owner, then the director is also the backup owner. If the owner chooses itself as the director, then a separate backup owner is chosen.
- Director—The unit that handles owner lookup requests from forwarders. When the owner receives a new connection, it chooses a director based on a hash of the source/destination IP address and ports, and sends a message to the director to register the new connection. If packets arrive at any unit other than the owner, the unit queries the director about which unit is the owner so it can forward the packets. A connection has only one director. If a director fails, the owner chooses a new director.

  As long as the director is not the same unit as the owner, then the director is also the backup owner (see above). If the owner chooses itself as the director, then a separate backup owner is chosen.
- Forwarder—A unit that forwards packets to the owner. If a forwarder receives a packet for a connection it does not own, it queries the director for the owner, and then establishes a flow to the owner for any other packets it receives for this connection. The director can also be a forwarder. Note that if a forwarder receives the SYN-ACK packet, it can derive the owner directly from a SYN cookie in the packet, so it does not need to query the director. (If you disable TCP sequence randomization, the SYN cookie is not used; a query to the director is required.) For short-lived flows such as DNS and ICMP, instead of querying, the forwarder immediately sends the packet to the director, which then sends them to the owner. A connection can have multiple forwarders; the most efficient throughput is achieved by a good load-balancing method where there are no forwarders and all packets of a connection are received by the owner.

Note We do not recommend disabling TCP sequence randomization when using clustering. There is a small chance that some TCP sessions won't be established, because the SYN/ACK packet might be dropped.

- Fragment Owner—For fragmented packets, cluster units that receive a fragment determine a fragment owner using a hash of the fragment source IP address, destination IP address, and the packet ID. All fragments are then forwarded to the fragment owner over the cluster control link. Fragments may be load-balanced to different cluster units, because only the first fragment includes the 5-tuple used in the switch load balance hash. Other fragments do not contain the source and destination ports and may be load-balanced to other cluster units. The fragment owner temporarily reassembles the packet so it can determine the director based on a hash of the source/destination IP address and ports. If it is a new connection, the fragment owner will register to be the connection owner. If it is an existing connection, the fragment owner forwards all fragments to the provided connection owner over the cluster control link. The connection owner will then reassemble all fragments.

When a connection uses Port Address Translation (PAT), then the PAT type (per-session or multi-session) influences which member of the cluster becomes the owner of a new connection:

- Per-session PAT—The owner is the unit that receives the initial packet in the connection.

  By default, TCP and DNS UDP traffic use per-session PAT.

- Multi-session PAT—The owner is always the control unit. If a multi-session PAT connection is initially received by a data unit, then the data unit forwards the connection to the control unit.

  By default, UDP (except for DNS UDP) and ICMP traffic use multi-session PAT, so these connections are always owned by the control unit.

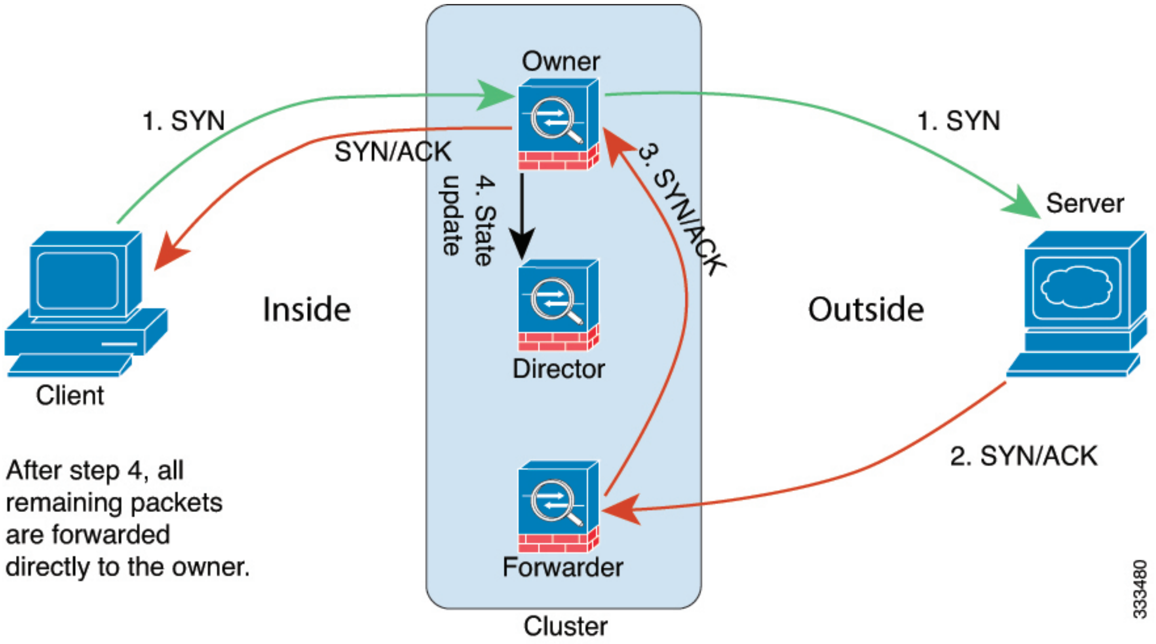
You can change the per-session PAT defaults for TCP and UDP so connections for these protocols are handled per-session or multi-session depending on the configuration. For ICMP, you cannot change from the default multi-session PAT. For more information about per-session PAT, see the firewall configuration guide.

New Connection Ownership

When a new connection is directed to a member of the cluster via load balancing, that unit owns both directions of the connection. If any connection packets arrive at a different unit, they are forwarded to the owner unit over the cluster control link. For best performance, proper external load balancing is required for both directions of a flow to arrive at the same unit, and for flows to be distributed evenly between units. If a reverse flow arrives at a different unit, it is redirected back to the original unit.

Sample Data Flow

The following example shows the establishment of a new connection.



1. The SYN packet originates from the client and is delivered to one ASA (based on the load balancing method), which becomes the owner. The owner creates a flow, encodes owner information into a SYN cookie, and forwards the packet to the server.
2. The SYN-ACK packet originates from the server and is delivered to a different ASA (based on the load balancing method). This ASA is the forwarder.
3. Because the forwarder does not own the connection, it decodes owner information from the SYN cookie, creates a forwarding flow to the owner, and forwards the SYN-ACK to the owner.
4. The owner sends a state update to the director, and forwards the SYN-ACK to the client.
5. The director receives the state update from the owner, creates a flow to the owner, and records the TCP state information as well as the owner. The director acts as the backup owner for the connection.
6. Any subsequent packets delivered to the forwarder will be forwarded to the owner.
7. If packets are delivered to any additional units, it will query the director for the owner and establish a flow.
8. Any state change for the flow results in a state update from the owner to the director.

Rebalancing New TCP Connections Across the Cluster

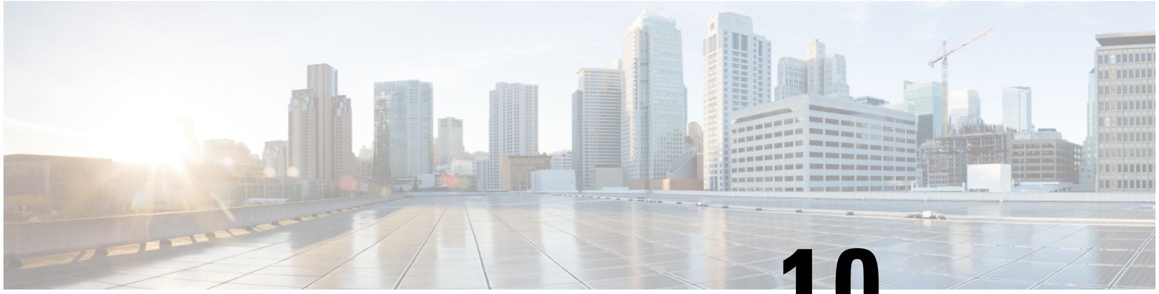
If the load balancing capabilities of the upstream or downstream routers result in unbalanced flow distribution, you can configure overloaded units to redirect new TCP flows to other units. No existing flows will be moved to other units.

History for ASA Clustering

Feature Name	Version	Feature Information
Enable and disable ASA cluster health monitoring per interface	9.4(1)	You can now enable or disable health monitoring per interface. Health monitoring is enabled by default on all port-channel, redundant, and single physical interfaces. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored. You might want to disable health monitoring of non-essential interfaces, for example, the management interface. We introduced the following command: **health-check monitor-interface**.
ASA clustering support for DHCP relay	9.4(1)	You can now configure DHCP relay on the ASA cluster. Client DHCP requests are load-balanced to the cluster members using a hash of the client MAC address. DHCP client and server functions are still not supported. We did not modify any commands.
SIP inspection support in ASA clustering	9.4(1)	You can now configure SIP inspection on the ASA cluster. A control flow can be created on any unit (due to load balancing), but its child data flows must reside on the same unit. TLS Proxy configuration is not supported. We introduced the following command: **show cluster service-policy**.
Inter-site deployment in transparent mode with the ASA cluster firewalling between inside networks	9.3(2)	You can now deploy a cluster in transparent mode between inside networks and the gateway router at each site (AKA East-West insertion), and extend the inside VLANs between sites. We recommend using Overlay Transport Virtualization (OTV), but you can use any method that ensures that the overlapping MAC Addresses and IP addresses of the gateway router do not leak between sites. Use a First Hop Redundancy Protocol (FHRP) such as HSRP to provide the same virtual MAC and IP addresses to the gateway routers.
BGP support for ASA clustering	9.3(1)	We added support for BGP with ASA clustering. We introduced the following command: **bgp router-id clusterpool**.
Support for cluster members at different geographical locations (inter-site) for transparent mode	9.2(1)	You can now place cluster members at different geographical locations when using Spanned EtherChannel mode in transparent firewall mode. Inter-site clustering with spanned EtherChannels in routed firewall mode is not supported. We did not modify any commands.
Static LACP port priority support for clustering	9.2(1)	Some switches do not support dynamic port priority with LACP (active and standby links). You can now disable dynamic port priority to provide better compatibility with spanned EtherChannels. You should also follow these guidelines: • Network elements on the cluster control link path should not verify the L4 checksum. Redirected traffic over the cluster control link does not have a correct L4 checksum. Switches that verify the L4 checksum could cause traffic to be dropped. • Port-channel bundling downtime should not exceed the configured keepalive interval. We introduced the following command: **clacp static-port-priority**.

Feature Name	Version	Feature Information
Support for 32 active links in a spanned EtherChannel	9.2(1)	ASA EtherChannels now support up to 16 active links. With *spanned* EtherChannels, that functionality is extended to support up to 32 active links across the cluster when used with two switches in a vPC and when you disable dynamic port priority. The switches must support EtherChannels with 16 active links, for example, the Cisco Nexus 7000 with F2-Series 10 Gigabit Ethernet Module. For switches in a VSS or vPC that support 8 active links, you can now configure 16 active links in the spanned EtherChannel (8 connected to each switch). Previously, the spanned EtherChannel only supported 8 active links and 8 standby links, even for use with a VSS/vPC. **Note** If you want to use more than 8 active links in a spanned EtherChannel, you cannot also have standby links; the support for 9 to 32 active links requires you to disable cLACP dynamic port priority that allows the use of standby links. We introduced the following command: **clacp static-port-priority**.
Support for 16 cluster members for the ASA 5585-X	9.2(1)	The ASA 5585-X now supports 16-unit clusters. We did not modify any commands.
ASA 5500-X support for clustering	9.1(4)	The ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, and ASA 5555-X now support 2-unit clusters. Clustering for 2 units is enabled by default in the base license; for the ASA 5512-X, you need the Security Plus license. We did not modify any commands.
Improved VSS and vPC support for health check monitoring	9.1(4)	If you configure the cluster control link as an EtherChannel (recommended), and it is connected to a VSS or vPC pair, you can now increase stability with health check monitoring. For some switches, such as the Cisco Nexus 5000, when one unit in the VSS/vPC is shutting down or booting up, EtherChannel member interfaces connected to that switch may appear to be Up to the ASA, but they are not passing traffic on the switch side. The ASA can be erroneously removed from the cluster if you set the ASA holdtime timeout to a low value (such as .8 seconds), and the ASA sends keepalive messages on one of these EtherChannel interfaces. When you enable the VSS/vPC health check feature, the ASA floods the keepalive messages on all EtherChannel interfaces in the cluster control link to ensure that at least one of the switches can receive them. We modified the following command: **health-check** [**vss-enabled**].
Support for cluster members at different geographical locations (inter-site); Individual Interface mode only	9.1(4)	You can now place cluster members at different geographical locations when using Individual Interface mode. We did not modify any commands.

Feature Name	Version	Feature Information
ASA Clustering for the ASA 5580 and 5585-X	9.0(1)	ASA Clustering lets you group up to 8 ASAs together as a single logical device. A cluster provides all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices. ASA clustering is supported for the ASA 5580 and the ASA 5585-X; all units in a cluster must be the same model with the same hardware specifications. See the configuration guide for a list of unsupported features when clustering is enabled. We introduced or modified the following commands: **channel-group**, **clacp system-mac**, **clear cluster info**, **clear configure cluster**, **cluster exec**, **cluster group**, **cluster interface-mode**, **cluster-interface**, **conn-rebalance, console-replicate**, **cluster master unit**, **cluster remove unit**, **debug cluster**, **debug lacp cluster**, **enable** (cluster group), **health-check**, **ip address**, **ipv6 address**, **key** (cluster group), **local-unit, mac-address** (interface), **mac-address pool**, **mtu cluster, port-channel span-cluster, priority** (cluster group), **prompt cluster-unit**, **show asp cluster counter**, **show asp table cluster chash-table**, **show cluster**, **show cluster info**, **show cluster user-identity**, **show lacp cluster**, and **show running-config cluster**.



CHAPTER 10

ASA Cluster for the Firepower 9300 Chassis

Clustering lets you group multiple Firepower 9300 chassis ASAs together as a single logical device. The Firepower 9300 chassis series includes the Firepower 9300. A cluster provides all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices.

Note The Firepower 9300 does not support a cluster across multiple chassis (inter-chassis); only intra-chassis clustering is supported.

Note Some features are not supported when using clustering. See Unsupported Features with Clustering, on page 386.

About Clustering on the Firepower 9300 Chassis

The cluster consists of multiple devices acting as a single logical unit. When you deploy a cluster on the Firepower 9300 chassis, it does the following:

- Creates a *cluster-control link* (by default, port-channel 48) for unit-to-unit communication.

  For intra-chassis clustering, this link utilizes the Firepower 9300 backplane for cluster communications.

- Creates the cluster bootstrap configuration within the application.

When you deploy the cluster, the chassis supervisor pushes a minimal bootstrap configuration to each unit that includes the cluster name, cluster control link interface, and other cluster settings. Some parts of the bootstrap configuration may be user-configurable within the application if you want to customize your clustering environment.

- Assigns data interfaces to the cluster as *Spanned* interfaces.

  For intra-chassis clustering, spanned interfaces are not limited to EtherChannels, like it is for inter-chassis clustering.The Firepower 9300 supervisor uses EtherChannel technology internally to load-balance traffic to multiple modules on a shared interface, so any data interface type works for Spanned mode.

  Note Individual interfaces are not supported, with the exception of a management interface.

- Assigns a management interface to all units in the cluster.

The following sections provide more detail about clustering concepts and implementation. See also Reference for Clustering, on page 386.

Bootstrap Configuration

When you deploy the cluster, the Firepower 9300 chassis supervisor pushes a minimal bootstrap configuration to each unit that includes the cluster name, cluster control link interface, and other cluster settings. Some parts of the bootstrap configuration are user-configurable if you want to customize your clustering environment.

Cluster Members

Cluster members work together to accomplish the sharing of the security policy and traffic flows.

One member of the cluster is the **control** unit. The control unit is determined automatically. All other members are **data** units.

You must perform all configuration on the control unit only; the configuration is then replicated to the data units.

Some features do not scale in a cluster, and the control unit handles all traffic for those features. See Centralized Features for Clustering, on page 387.

Master and Slave Unit Roles

One member of the cluster is the master unit. The master unit is determined automatically. All other members are slave units.

You must perform all configuration on the master unit only; the configuration is then replicated to the slave units.

Some features do not scale in a cluster, and the master unit handles all traffic for those features. See Centralized Features for Clustering, on page 387.

Cluster Control Link

The cluster-control link is an EtherChannel (port-channel 48) for unit-to-unit communication. For intra-chassis clustering, this link utilizes the Firepower 9300 backplane for cluster communications.

Cluster control link traffic includes both control and data traffic.

Control traffic includes:

- Control unit election.
- Configuration replication.
- Health monitoring.

Data traffic includes:

- State replication.
- Connection ownership queries and data packet forwarding.

Cluster Control Link Network

The Firepower 9300 chassis auto-generates the cluster control link interface IP address for each unit based on the chassis ID and slot ID: 127.2.*chassis_id*.*slot_id*. You cannot set this IP address manually, either in FXOS or within the application. The cluster control link network cannot include any routers between units; only Layer 2 switching is allowed.

Cluster Interfaces

For intra-chassis clustering, you can assign both physical interfaces or EtherChannels (also known as port channels) to the cluster. Interfaces assigned to the cluster are Spanned interfaces that load-balance traffic across all members of the cluster.

Individual interfaces are not supported, with the exception of a management interface.

Connecting to a VSS or vPC

We recommend connecting EtherChannels to a VSS or vPC to provide redundancy for your interfaces.

Configuration Replication

All units in the cluster share a single configuration. You can only make configuration changes on the control unit, and changes are automatically synced to all other units in the cluster.

ASA Cluster Management

One of the benefits of using ASA clustering is the ease of management. This section describes how to manage the cluster.

Management Interface

You must assign a Management type interface to the cluster. This interface is a special individual interface as opposed to a Spanned interface. The management interface lets you connect directly to each unit.

The Main cluster IP address is a fixed address for the cluster that always belongs to the current control unit. You also configure a range of addresses so that each unit, including the current control unit, can use a Local address from the range. The Main cluster IP address provides consistent management access to an address; when a control unit changes, the Main cluster IP address moves to the new control unit, so management of the cluster continues seamlessly.

For example, you can manage the cluster by connecting to the Main cluster IP address, which is always attached to the current control unit. To manage an individual member, you can connect to the Local IP address.

For outbound management traffic such as TFTP or syslog, each unit, including the control unit, uses the Local IP address to connect to the server.

Control Unit Management Vs. Data Unit Management

All management and monitoring can take place on the control unit. From the control unit, you can check runtime statistics, resource usage, or other monitoring information of all units. You can also issue a command to all units in the cluster, and replicate the console messages from data units to the control unit.

You can monitor data units directly if desired. Although also available from the control unit, you can perform file management on data units (including backing up the configuration and updating images). The following functions are not available from the control unit:

- Monitoring per-unit cluster-specific statistics.
- Syslog monitoring per unit (except for syslogs sent to the console when console replication is enabled).
- SNMP
- NetFlow

RSA Key Replication

When you create an RSA key on the control unit, the key is replicated to all data units. If you have an SSH session to the Main cluster IP address, you will be disconnected if the control unit fails. The new control unit uses the same key for SSH connections, so that you do not need to update the cached SSH host key when you reconnect to the new control unit.

ASDM Connection Certificate IP Address Mismatch

By default, a self-signed certificate is used for the ASDM connection based on the Local IP address. If you connect to the Main cluster IP address using ASDM, then a warning message about a mismatched IP address might appear because the certificate uses the Local IP address, and not the Main cluster IP address. You can ignore the message and establish the ASDM connection. However, to avoid this type of warning, you can enroll a certificate that contains the Main cluster IP address and all the Local IP addresses from the IP address pool. You can then use this certificate for each cluster member. See https://www.cisco.com/c/en/us/td/docs/security/asdm/identity-cert/cert-install.html for more information.

Requirements and Prerequisites for Clustering on the Firepower 9300 Chassis

Maximum Clustering Units Per Model

- Firepower 9300—16 modules. For example, you can use 1 module in 16 chassis, or 2 modules in 8 chassis, or any combination that provides a maximum of 16 modules.

Switch Requirements

- Be sure to complete the switch configuration and successfully connect all the EtherChannels from the chassis to the switch(es) before you configure clustering on the Firepower 9300 chassis.
- For supported switch characteristics, see Cisco FXOS Compatibility.

Licenses for Clustering on the Firepower 9300 Chassis

The clustering feature itself does not require any licenses. To use Strong Encryption and other optional licenses, you can only request licenses on the control unit; the licenses are aggregated with the data units. If you have licenses on multiple units, they combine into a single running ASA cluster license. License configuration completed on the control unit is not replicated to the data units. You can only configure separate license entitlements on data units if you disable clustering, configure the licensing, and then re-enable clustering.

Note To use ASDM and other strong encryption features, after you deploy the cluster you must enable the Strong Encryption (3DES) license on the control unit using the ASA CLI. This license is inherited by the data units; you do not need to configure this license separately on each unit. The Strong Encryption (3DES) license is not available with any type of evaluation license.

Note If the control unit fails, and does not rejoin within 30 days (the licensing grace period), then the inherited licenses disappear. You must then manually configure the missing licenses on the new control unit.

Clustering Guidelines and Limitations

- When significant topology changes occur (such as adding or removing an EtherChannel interface, enabling or disabling an interface on the Firepower 9300 chassis or the switch, adding an additional switch to form a VSS or vPC) you should disable the health check feature, and also disable interface monitoring for the disabled interfaces . When the topology change is complete, and the configuration change is synced to all units, you can re-enable the health check feature.

- When adding a unit to an existing cluster, or when reloading a unit, there will be a temporary, limited packet/connection drop; this is expected behavior. In some cases, the dropped packets can hang connections; for example, dropping a FIN/ACK packet for an FTP connection will make the FTP client hang. In this case, you need to reestablish the FTP connection.
- If you use a Windows 2003 server connected to a Spanned EtherChannel interface, when the syslog server port is down, and the server does not throttle ICMP error messages, then large numbers of ICMP messages are sent back to the cluster. These messages can result in some units of the cluster experiencing high CPU, which can affect performance. We recommend that you throttle ICMP error messages.
- We recommend connecting EtherChannels to a VSS or vPC for redundancy.
- Within a chassis, you cannot cluster some security modules and run other security modules in standalone mode; you must include all security modules in the cluster.

Defaults

- The cluster health check feature is enabled by default with the holdtime of 3 seconds. Interface health monitoring is enabled on all interfaces by default.
- Connection rebalancing is disabled by default. If you enable connection rebalancing, the default time between load information exchanges is 5 seconds.

Configure Clustering on the Firepower 9300 Chassis

You can easily deploy the cluster from the Firepower 9300 chassis supervisor. All initial configuration is automatically generated for each unit. This section describes the default bootstrap configuration and optional customization you can perform on the ASA. This section also describes how to manage cluster members from within the ASA. You can also manage cluster membership from the Firepower 9300 chassis. See the Firepower 9300 chassis documentation for more information.

Procedure

Step 1 FXOS: Add an ASA Cluster, on page 362

Step 2 ASA: Change the Firewall Mode and Context Mode, on page 369

Step 3 ASA: Configure Data Interfaces, on page 370

Step 4 ASA: Customize the Cluster Configuration, on page 372

Step 5 ASA: Manage Cluster Members, on page 377

FXOS: Add an ASA Cluster

You can add a single Firepower 9300 chassis as an intra-chassis cluster.

Create an ASA Cluster

Set the scope to the image version.

You can easily deploy the cluster from the Firepower 9300 chassis supervisor. All initial configuration is automatically generated for each unit.

In a Firepower 9300 chassis, you must enable clustering for all 3 module slots, even if you do not have a module installed. If you do not configure all 3 modules, the cluster will not come up.

For multiple context mode, you must first deploy the logical device, and then enable multiple context mode in the ASA application.

To change the ASA to transparent firewall mode, complete the initial deployment, and then change the firewall mode within the ASA CLI.

When you deploy a cluster, the Firepower 9300 chassis supervisor configures each ASA application with the following bootstrap configuration. You can later modify parts of the bootstrap configuration from the ASA, if desired (shown in **Bold** text).

```
interface Port-channel48
   description Clustering Interface
cluster group <service_type_name>
   key <secret>
   local-unit unit-<chassis#-module#>

   cluster-interface port-channel48 ip 127.2.<chassis#>.<module#> 255.255.255.0
   priority <auto>
   health-check holdtime 3
   health-check data-interface auto-rejoin 3 5 2
   health-check cluster-interface auto-rejoin unlimited 5 1
   enable

ip local pool cluster_ipv4_pool <ip_address>-<ip_address> mask <mask>

interface <management_ifc>
   management-only individual
   nameif management
   security-level 0
   ip address <ip_address> <mask> cluster-pool cluster_ipv4_pool
   no shutdown

http server enable
http 0.0.0.0 0.0.0.0 management
route management <management_host_ip> <mask> <gateway_ip> 1
```

Note The **local-unit** name can only be changed if you disable clustering.

Before you begin

- Download the application image you want to use for the logical device from Cisco.com, and then upload that image to the Firepower 9300 chassis.
- Gather the following information:
  - Management interface ID, IP address, and network mask
  - Gateway IP address

Procedure

Step 1 Configure interfaces.

a) Add at least one Data type interface or EtherChannel (also known as a port-channel) before you deploy the cluster. See Add an EtherChannel (Port Channel), on page 136 or Configure a Physical Interface, on page 135.

b) Add a Management type interface or EtherChannel. See Add an EtherChannel (Port Channel), on page 136 or Configure a Physical Interface, on page 135.

The management interface is required. Note that this management interface is not the same as the chassis management interface that is used only for chassis management (in FXOS, you might see the chassis management interface displayed as MGMT, management0, or other similar names).

Step 2 Enter security services mode.

scope ssa

Example:

```
Firepower# scope ssa
Firepower /ssa #
```

Step 3 Set the application instance image version.

a) View available images. Note the Version number that you want to use.

show app

Example:

```
Firepower /ssa # show app
    Name       Version         Author     Supported Deploy Types CSP Type    Is Default
 App
    ---------- --------------- ---------- ---------------------- -----------
--------------
    asa        9.9.1           cisco      Native                 Application No
    asa        9.10.1          cisco      Native                 Application Yes
    ftd        6.2.3           cisco      Native                 Application Yes
```

b) Set the scope to the image version.

scope app asa *application_version*

Example:

```
Firepower /ssa # scope app asa 9.10.1
Firepower /ssa/app #
```

c) Set this version as the default.

set-default

Example:

```
Firepower /ssa/app # set-default
Firepower /ssa/app* #
```

d) Exit to ssa mode.

exit

Example:

```
Firepower /ssa/app* # exit
Firepower /ssa* #
```

Example:

```
Firepower /ssa # scope app asa 9.12.1
Firepower /ssa/app # set-default
Firepower /ssa/app* # exit
Firepower /ssa* #
```

Step 4 Create the cluster.

enter logical-device *device_name* **asa** *slots* **clustered**

- *device_name*—Used by the Firepower 9300 chassis supervisor to configure clustering settings and assign interfaces; it is not the cluster name used in the security module configuration. You must specify all three security modules, even if you have not yet installed the hardware.

- *slots*—Assigns the chassis modules to the cluster. For the Firepower 4100, specify **1**. For the Firepower 9300, specify **1,2,3**. You must enable clustering for all 3 module slots in a Firepower 9300 chassis, even if you do not have a module installed. If you do not configure all 3 modules, the cluster will not come up.

Example:

```
Firepower /ssa # enter logical-device ASA1 asa 1,2,3 clustered
Firepower /ssa/logical-device* #
```

Step 5 Configure the cluster bootstrap parameters.

These settings are meant for initial deployment only, or for disaster recovery. For normal operation, you can later change most values in the application CLI configuration.

a) Create the cluster bootstrap object.

enter cluster-bootstrap

Example:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* #
```

b) Configure an authentication key for control traffic on the cluster control link.

set key

Example:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set key
Key: diamonddogs
```

You are prompted to enter the shared secret.

The shared secret is an ASCII string from 1 to 63 characters. The shared secret is used to generate the key. This option does not affect datapath traffic, including connection state update and forwarded packets, which are always sent in the clear.

c) Set the cluster interface mode.

set mode spanned-etherchannel

Spanned EtherChannel mode is the only supported mode.

Example:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* #
```

d) Set the cluster group name in the security module configuration.

set service-type *cluster_name*

The name must be an ASCII string from 1 to 38 characters.

Example:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set service-type cluster1
Firepower /ssa/logical-device/cluster-bootstrap* #
```

e) Configure the management IP address information.

This information is used to configure a management interface in the security module configuration.

1. Configure a pool of Local IP addresses, one of which will be assigned to each cluster unit for the interface.

   set ipv4 pool *start_ip end_ip*

   set ipv6 pool *start_ip end_ip*

   Include at least as many addresses as there are units in the cluster. Note that for the Firepower 9300, you must include 3 addresses per chassis, even if you do not have all module slots filled. If you plan to expand the cluster, include additional addresses. The Virtual IP address (known as the Main cluster IP address) that belongs to the current control unit is *not* a part of this pool; be sure to reserve an IP address on the same network for the Main cluster IP address. You can use IPv4 and/or IPv6 addresses.

2. Configure the Main cluster IP address for the management interface.

   set virtual ipv4 *ip_address* **mask** *mask*

   set virtual ipv6 *ip_address* **prefix-length** *prefix*

   This IP address must be on the same network as the cluster pool addresses, but not be part of the pool.

3. Enter the network gateway address.

   set ipv4 gateway *ip_address*

set ipv6 gateway *ip_address*

Example:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv4 gateway 10.1.1.254
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv4 pool 10.1.1.11 10.1.1.27
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv6 gateway 2001:DB8::AA
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv6 pool 2001:DB8::11 2001:DB8::27
Firepower /ssa/logical-device/cluster-bootstrap* # set virtual ipv4 10.1.1.1 mask
255.255.255.0
Firepower /ssa/logical-device/cluster-bootstrap* # set virtual ipv6 2001:DB8::1
prefix-length 64
```

f) Exit the cluster bootstrap mode.

exit

Example:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* # set chassis-id 1
Firepower /ssa/logical-device/cluster-bootstrap* # set key
  Key: f@arscape
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* # set service-type cluster1
Firepower /ssa/logical-device/cluster-bootstrap* # exit
Firepower /ssa/logical-device/* #
```

Step 6 Configure the management bootstrap parameters.

These settings are meant for initial deployment only, or for disaster recovery. For normal operation, you can later change most values in the application CLI configuration.

a) Create the management bootstrap object.

enter mgmt-bootstrap asa

Example:

```
Firepower /ssa/logical-device* # enter mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

b) Specify the admin password.

create bootstrap-key-secret PASSWORD

set value

Enter a value: *password*

Confirm the value: *password*

exit

Example:

The pre-configured ASA admin user is useful for password recovery; if you have FXOS access, you can reset the admin user password if you forget it.

Example:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Enter a value: floppylampshade
Confirm the value: floppylampshade
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

c) Exit the management bootstrap mode.

exit

Example:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* #
```

Step 7 Save the configuration.

commit-buffer

The chassis deploys the logical device by downloading the specified software version and pushing the bootstrap configuration and management interface settings to the application instance. Check the status of the deployment using the **show app-instance** command. The application instance is running and ready to use when the **Admin State** is **Enabled** and the **Oper State** is **Online**.

Example:

```
Firepower /ssa/logical-device* # commit-buffer
Firepower /ssa/logical-device # exit
Firepower /ssa # show app-instance
App Name   Identifier Slot ID    Admin State Oper State      Running Version Startup Version
 Deploy Type Profile Name Cluster State   Cluster Role
---------- ---------- ---------- ----------- ---------------- --------------- ---------------
 ----------- ------------ --------------- ------------
ftd        cluster1   1          Enabled     Online           6.4.0.49        6.4.0.49
    Native                   In Cluster      Slave
ftd        cluster1   2          Enabled     Online           6.4.0.49        6.4.0.49
    Native                   In Cluster      Master
ftd        cluster1   3          Disabled    Not Available                    6.4.0.49
    Native                   Not Applicable  None
```

Step 8 Connect to the control unit ASA to customize your clustering configuration.

Example

For chassis 1:

```
scope eth-uplink
  scope fabric a
    enter port-channel 1
      set port-type data
      enable
      enter member-port Ethernet1/1
        exit
      enter member-port Ethernet1/2
```

```
        exit
      exit
    enter port-channel 2
      set port-type data
      enable
      enter member-port Ethernet1/3
        exit
      enter member-port Ethernet1/4
        exit
      exit
    enter port-channel 3
      set port-type data
      enable
      enter member-port Ethernet1/5
        exit
      enter member-port Ethernet1/6
        exit
      exit
    enter port-channel 4
      set port-type mgmt
      enable
      enter member-port Ethernet2/1
        exit
      enter member-port Ethernet2/2
        exit
      exit

    exit
  exit
commit-buffer

scope ssa
  enter logical-device ASA1 asa "1,2,3" clustered
    enter cluster-bootstrap
      set chassis-id 1
      set ipv4 gateway 10.1.1.254
      set ipv4 pool 10.1.1.11 10.1.1.27
      set ipv6 gateway 2001:DB8::AA
      set ipv6 pool 2001:DB8::11 2001:DB8::27
      set key
      Key: f@arscape
      set mode spanned-etherchannel
      set service-type cluster1
      set virtual ipv4 10.1.1.1 mask 255.255.255.0
      set virtual ipv6 2001:DB8::1 prefix-length 64
      exit
    exit
  scope app asa 9.5.2.1
    set-default
    exit
  commit-buffer
```

ASA: Change the Firewall Mode and Context Mode

By default, the FXOS chassis deploys a cluster in routed or transparent firewall mode, and single context mode.

- Change the firewall mode— To change the mode after you depoy, change the mode on the control unit; the mode is automatically changed on all data units to match. See Set the Firewall Mode, on page 157. In multiple context mode, you set the firewall mode per context.

- Change to multiple context mode—To change to multiple context mode after you deploy, change the mode on the control unit; the mode is automatically changed on all data units to match. See Enable Multiple Context Mode, on page 186.

ASA: Configure Data Interfaces

This procedure configures basic parameters for each data interface that you assigned to the cluster when you deployed it in FXOS. For inter-chassis clustering, data interfaces are always Spanned EtherChannel interfaces.

Note The management interface was pre-configured when you deployed the cluster. You can also change the management interface parameters in ASA, but this procedure focuses on data interfaces. The management interface is an individual interface, as opposed to a Spanned interface. See Management Interface, on page 360 for more information.

Before you begin

- For multiple context mode, start this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.
- For transparent mode, configure the bridge group. See Configure the Bridge Virtual Interface (BVI), on page 458.
- When using Spanned EtherChannels for inter-chassis clustering, the port-channel interface will not come up until clustering is fully enabled. This requirement prevents traffic from being forwarded to a unit that is not an active unit in the cluster.

Procedure

Step 1 Specify the interface ID.

interface *id*

Refer to the FXOS chassis for the interfaces assigned to this cluster. The interface ID can be:

- **port-channel** *integer*
- **ethernet** *slot/port*

Example:

```
ciscoasa(config)# interface port-channel 1
```

Step 2 Enable the interface:

no shutdown

Step 3 (Optional) If you are creating VLAN subinterfaces on this interface, do so now.

Example:

```
ciscoasa(config)# interface port-channel 1.10
ciscoasa(config-if)# vlan 10
```

The rest of this procedure applies to the subinterfaces.

Step 4 (Multiple Context Mode) Allocate the interface to a context, then changeto the context and enter interface mode.

Example:

```
ciscoasa(config)# context admin
ciscoasa(config)# allocate-interface port-channel1
ciscoasa(config)# changeto context admin
ciscoasa(config-if)# interface port-channel 1
```

For multiple context mode, the rest of the interface configuration occurs within each context.

Step 5 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif inside
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value.

Step 6 Perform one of the following, depending on the firewall mode.

- Routed Mode—Set the IPv4 and/or IPv6 address:

  (IPv4)

  ip address *ip_address* [*mask*]

  (IPv6)

  ipv6 address *ipv6-prefix*/*prefix-length*

  Example:

  ```
  ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
  ciscoasa(config-if)# ipv6 address 2001:DB8::1001/32
  ```

  DHCP, PPPoE, and IPv6 autoconfig are not supported.

- Transparent Mode—Assign the interface to a bridge group:

  bridge-group *number*

  Example:

  ```
  ciscoasa(config-if)# bridge-group 1
  ```

Where *number* is an integer between 1 and 100. You can assign up to 4 interfaces to a bridge group. You cannot assign the same interface to more than one bridge group. Note that the BVI configuration includes the IP address.

Step 7 Set the security level:

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 50
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 8 (Inter-chassis clustering) Configure a global MAC address for a Spanned EtherChannel to avoid potential network connectivity problems.

mac-address *mac_address*

- *mac_address*—The MAC address is in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE is entered as 000C.F142.4CDE. The first two bytes of a manual MAC address cannot be A2 if you also want to use auto-generated MAC addresses.

With a manually-configured MAC address, the MAC address stays with the current control unit. If you do not configure a MAC address, then if the control unit changes, the new control unit uses a new MAC address for the interface, which can cause a temporary network outage.

In multiple context mode, if you share an interface between contexts, you should instead enable auto-generation of MAC addresses so you do not need to set the MAC address manually. Note that you must manually configure the MAC address using this command for *non-shared* interfaces.

Example:

```
ciscoasa(config-if)# mac-address 000C.F142.4CDE
```

ASA: Customize the Cluster Configuration

If you want to change bootstrap settings after you deploy the cluster or configure additional options, such as clustering health monitoring, TCP connection replication delay, flow mobility, and other optimizations, you can do so on the control unit.

Configure Basic ASA Cluster Parameters

You can customize cluster settings on the control unit.

Before you begin

- For multiple context mode, complete this procedure in the system execution space on the control unit. To change from the context to the system execution space, enter the **changeto system** command.

- The local-unit name and several other options can only be set on the FXOS chassis, or they can only be changed on the ASA if you disable clustering, so they are not included in the following procedure.

Procedure

Step 1 Confirm that this unit is the control unit:

show cluster info

Example:

```
asa(config)# show cluster info
Cluster cluster1: On
    Interface mode: spanned
    This is "unit-1-2" in state MASTER
        ID        : 2
        Version   : 9.5(2)
        Serial No.: FCH183770GD
        CCL IP    : 127.2.1.2
        CCL MAC   : 0015.c500.019f
        Last join : 01:18:34 UTC Nov 4 2015
        Last leave: N/A
Other members in the cluster:
    Unit "unit-1-3" in state SLAVE
        ID        : 4
        Version   : 9.5(2)
        Serial No.: FCH19057ML0
        CCL IP    : 127.2.1.3
        CCL MAC   : 0015.c500.018f
        Last join : 20:29:57 UTC Nov 4 2015
        Last leave: 20:24:55 UTC Nov 4 2015
    Unit "unit-1-1" in state SLAVE
        ID        : 1
        Version   : 9.5(2)
        Serial No.: FCH19057ML0
        CCL IP    : 127.2.1.1
        CCL MAC   : 0015.c500.017f
        Last join : 20:20:53 UTC Nov 4 2015
        Last leave: 20:18:15 UTC Nov 4 2015
```

If a different unit is the control unit, exit the connection and connect to the correct unit. See the Cisco ASA for Firepower 4100 Quick Start Guide or the Cisco ASA for Firepower 9300 Quick Start Guide for information about accessing the ASA console.

Step 2 Specify the maximum transmission unit for the cluster control link interface to be at least 100 bytes higher than the highest MTU of the data interfaces.

mtu cluster *bytes*

Example:

```
ciscoasa(config)# mtu cluster 9000
```

We suggest setting the cluster control link MTU to the maximum; the minimum value is 1400 bytes. Because the cluster control link traffic includes data packet forwarding, the cluster control link needs to accommodate the entire size of a data packet plus cluster traffic overhead. For example, because the maximum MTU is 9000, then the highest data interface MTU can be 8900, while the cluster control link can be set to 9000.

Step 3 Enter cluster configuration mode:

cluster group *name*

Step 4 (Optional) Enable console replication from data units to the control unit:

console-replicate

This feature is disabled by default. The ASA prints out some messages directly to the console for certain critical events. If you enable console replication, data units send the console messages to the control unit so that you only need to monitor one console port for the cluster.

Step 5 (Optional) Disable dynamic port priority in LACP.

clacp static-port-priority

Some switches do not support dynamic port priority, so this command improves switch compatibility. Moreover, it enables support of more than 8 active spanned EtherChannel members, up to 32 members. Without this command, only 8 active members and 8 standby members are supported. If you enable this command, then you cannot use any standby members; all members are active.

Configure Health Monitoring

This procedure configures unit and interface health monitoring.

You might want to disable health monitoring of non-essential interfaces, for example, the management interface. You can monitor any port-channel ID or single physical interface ID. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored.

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Step 2 Customize the cluster unit health check feature:

health-check [**holdtime** *timeout*]

The **holdime** determines the amount of time between unit keepalive status messages, between .8 and 45 seconds; The default is 3 seconds.

To determine unit health, the ASA cluster units send keepalive messages on the cluster control link to other units. If a unit does not receive any keepalive messages from a peer unit within the holdtime period, the peer unit is considered unresponsive or dead.

When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the ASA, Firepower 9300 chassis, or the switch, or adding an additional switch to form a VSS or vPC) you should disable the health check feature and also disable interface monitoring for the disabled interfaces (**no health-check monitor-interface**). When the topology change is complete, and the configuration change is synced to all units, you can re-enable the health check feature.

Example:

```
ciscoasa(cfg-cluster)# health-check holdtime 5
```

Step 3 Disable the interface health check on an interface:

no health-check monitor-interface [*interface_id*]

The interface health check monitors for link failures. If all physical ports for a given logical interface fail on a particular unit, but there are active ports under the same logical interface on other units, then the unit is removed from the cluster. The amount of time before the ASA removes a member from the cluster depends on the type of interface and whether the unit is an established member or is joining the cluster.

Health check is enabled by default for all interfaces. You can disable it per interface using the **no** form of this command. You might want to disable health monitoring of non-essential interfaces, for example, the management interface. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored.

When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the ASA, Firepower 9300 chassis, or the switch, or adding an additional switch to form a VSS or vPC) you should disable the health check feature (**no health-check**) and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all units, you can re-enable the health check feature.

Example:

```
ciscoasa(cfg-cluster)# no health-check monitor-interface port-channel1
```

Step 4 Configure the chassis health check interval:

app-agent heartbeat [**interval** *ms*] [**retry-count** *number*]

- **interval** *ms*—Set the amount of time between heartbeats, between 300 and 6000 ms, in multiples of 100. The default is 1000 ms.
- **retry-count** *number*—Set the number of retries, between 1 and 30. The default is 3 retries.

The ASA checks whether it can communicate over the backplane with the host Firepower chassis.

Example:

```
ciscoasa(cfg-cluster)# app-agent heartbeat interval 300
```

Configure Connection Rebalancing

You can configure connection rebalancing.

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Step 2 (Optional) Enable connection rebalancing for TCP traffic:

conn-rebalance [**frequency** *seconds*]

Example:

```
ciscoasa(cfg-cluster)# conn-rebalance frequency 60
```

This command is disabled by default. If enabled, ASAs exchange load information periodically, and offload new connections from more loaded devices to less loaded devices. The frequency, between 1 and 360 seconds, specifies how often the load information is exchanged. The default is 5 seconds.

FXOS: Remove a Cluster Unit

The following sections describe how to remove units temporarily or permanently from the cluster.

Temporary Removal

A cluster unit will be automatically removed from the cluster due to a hardware or network failure, for example. This removal is temporary until the conditions are rectified, and it can rejoin the cluster. You can also manually disable clustering.

To check whether a device is currently in the cluster, check the cluster status within the application using the **show cluster info** command:

```
ciscoasa# show cluster info
Clustering is not enabled
```

- Disable clustering in the application—You can disable clustering using the application CLI. Enter the **cluster remove unit** *name* command to remove any unit other than the one you are logged into. The bootstrap configuration remains intact, as well as the last configuration synced from the control unit, so you can later re-add the unit without losing your configuration. If you enter this command on a data unit to remove the control unit, a new control unit is elected.

  When a device becomes inactive, all data interfaces are shut down; only the Management interface can send and receive traffic. To resume traffic flow, re-enable clustering. The Management interface remains up using the IP address the unit received from the bootstrap configuration. However if you reload, and the unit is still inactive in the cluster (for example, you saved the configuration with clustering disabled), the Management interface is disabled.

  To reenable clustering, on the ASA enter **cluster group** *name* and then **enable**.

- Disable the application instance—At the FXOS CLI, see the following example:

  ```
  Firepower-chassis# scope ssa
  Firepower-chassis /ssa # scope slot 1
  Firepower-chassis /ssa/slot # scope app-instance asa asa1
  Firepower-chassis /ssa/slot/app-instance # disable
  Firepower-chassis /ssa/slot/app-instance* # commit-buffer
  Firepower-chassis /ssa/slot/app-instance #
  ```

  To reenable:

```
Firepower-chassis /ssa/slot/app-instance # enable
Firepower-chassis /ssa/slot/app-instance* # commit-buffer
Firepower-chassis /ssa/slot/app-instance #
```

- Shut down the security module/engine—At the FXOS CLI, see the following example:

  ```
  Firepower-chassis# scope service-profile server 1/1
  Firepower-chassis /org/service-profile # power down soft-shut-down
  Firepower-chassis /org/service-profile* # commit-buffer
  Firepower-chassis /org/service-profile #
  ```

  To power up:

  ```
  Firepower-chassis /org/service-profile # power up
  Firepower-chassis /org/service-profile* # commit-buffer
  Firepower-chassis /org/service-profile #
  ```

- Shut down the chassis—At the FXOS CLI, see the following example:

  ```
  Firepower-chassis# scope chassis 1
  Firepower-chassis /chassis # shutdown no-prompt
  ```

Permanent Removal

You can permanently remove a cluster member using the following methods.

- Delete the logical device—At the FXOS CLI, see the following example:

  ```
  Firepower-chassis# scope ssa
  Firepower-chassis /ssa # delete logical-device cluster1
  Firepower-chassis /ssa* # commit-buffer
  Firepower-chassis /ssa #
  ```

- Remove the chassis or security module from service—If you remove a device from service, you can add replacement hardware as a new member of the cluster.

ASA: Manage Cluster Members

After you deploy the cluster, you can change the configuration and manage cluster members.

Become an Inactive Member

To become an inactive member of the cluster, disable clustering on the unit while leaving the clustering configuration intact.

Note When an ASA becomes inactive (either manually or through a health check failure), all data interfaces are shut down; only the management-only interface can send and receive traffic. To resume traffic flow, re-enable clustering; or you can remove the unit altogether from the cluster. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster (for example, you saved the configuration with clustering disabled), then the management interface is disabled. You must use the console port for any further configuration.

Before you begin

- You must use the console port; you cannot enable or disable clustering from a remote CLI connection.
- For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Step 1 Enter cluster configuration mode:

cluster group *name*

Example:

```
ciscoasa(config)# cluster group pod1
```

Step 2 Disable clustering:

no enable

If this unit was the control unit, a new control election takes place, and a different member becomes the control unit.

The cluster configuration is maintained, so that you can enable clustering again later.

Deactivate a Unit

To deactivate a member other than the unit you are logged into, perform the following steps.

Note When an ASA becomes inactive, all data interfaces are shut down; only the management-only interface can send and receive traffic. To resume traffic flow, re-enable clustering. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster (for example, if you saved the configuration with clustering disabled), the management interface is disabled. You must use the console port for any further configuration.

Before you begin

For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Remove the unit from the cluster.

cluster remove unit *unit_name*

The bootstrap configuration remains intact, as well as the last configuration synched from the control unit, so that you can later re-add the unit without losing your configuration. If you enter this command on a data unit to remove the control unit, a new control unit is elected.

To view member names, enter **cluster remove unit ?**, or enter the **show cluster info** command.

Example:

```
ciscoasa(config)# cluster remove unit ?

Current active units in the cluster:
asa2

ciscoasa(config)# cluster remove unit asa2
WARNING: Clustering will be disabled on unit asa2. To bring it back
to the cluster please logon to that unit and re-enable clustering
```

Rejoin the Cluster

If a unit was removed from the cluster, for example for a failed interface or if you manually deactivated a member, you must manually rejoin the cluster.

Before you begin

- You must use the console port to reenable clustering. Other interfaces are shut down.
- For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.
- Make sure the failure is resolved before you try to rejoin the cluster.

Procedure

Step 1 At the console, enter cluster configuration mode:

cluster group *name*

Example:

```
ciscoasa(config)# cluster group pod1
```

Step 2 Enable clustering.

enable

Change the Control Unit

Caution The best method to change the control unit is to disable clustering on the control unit, wait for a new control election, and then re-enable clustering. If you must specify the exact unit you want to become the control unit, use the procedure in this section. Note, however, that for centralized features, if you force a control unit change using this procedure, then all connections are dropped, and you have to re-establish the connections on the new control unit.

To change the control unit, perform the following steps.

Before you begin

For multiple context mode, perform this procedure in the system execution space. If you are not already in the System configuration mode, enter the **changeto system** command.

Procedure

Set a new unit as the control unit:

cluster master unit *unit_name*

Example:

```
ciscoasa(config)# cluster master unit asa2
```

You will need to reconnect to the Main cluster IP address.

To view member names, enter **cluster master unit ?** (to see all names except the current unit), or enter the **show cluster info** command.

Execute a Command Cluster-Wide

To send a command to all members in the cluster, or to a specific member, perform the following steps. Sending a **show** command to all members collects all output and displays it on the console of the current unit. (Note that alternatively there are show commands that you can enter on the control unit to view cluster-wide statistics.) Other commands, such as **capture** and **copy**, can also take advantage of cluster-wide execution.

Procedure

Send a command to all members, or if you specify the unit name, a specific member:

cluster exec [unit *unit_name*] *command*

Example:

```
ciscoasa# cluster exec show xlate
```

To view member names, enter **cluster exec unit ?** (to see all names except the current unit), or enter the **show cluster info** command.

Examples

To copy the same capture file from all units in the cluster at the same time to a TFTP server, enter the following command on the control unit:

```
ciscoasa# cluster exec copy /pcap capture: tftp://10.1.1.56/capture1.pcap
```

Multiple PCAP files, one from each unit, are copied to the TFTP server. The destination capture file name is automatically attached with the unit name, such as capture1_asa1.pcap, capture1_asa2.pcap, and so on. In this example, asa1 and asa2 are cluster unit names.

The following sample output for the **cluster exec show memory** command shows memory information for each member in the cluster:

```
ciscoasa# cluster exec show memory
unit-1-1(LOCAL):******************************************************
Free memory:      108724634538 bytes (92%)
Used memory:        9410087158 bytes ( 8%)
-------------     ------------------
Total memory:     118111600640 bytes (100%)


unit-1-3:*************************************************************
Free memory:      108749922170 bytes (92%)
Used memory:        9371097334 bytes ( 8%)
-------------     ------------------
Total memory:     118111600640 bytes (100%)


unit-1-2:*************************************************************
Free memory:      108426753537 bytes (92%)
Used memory:        9697869087 bytes ( 8%)
-------------     ------------------
Total memory:     118111600640 bytes (100%)
```

ASA: Monitoring the ASA Cluster on the Firepower 9300 chassis

You can monitor and troubleshoot cluster status and connections.

Monitoring Cluster Status

See the following commands for monitoring cluster status:

- **show cluster info [health], show cluster chassis info**

  With no keywords, the **show cluster info** command shows the status of all members of the cluster.

  The **show cluster info health** command shows the current health of interfaces, units, and the cluster overall.

  See the following output for the **show cluster info** command:

```
asa(config)# show cluster info
Cluster cluster1: On
    Interface mode: spanned
    This is "unit-1-2" in state MASTER
        ID        : 2
        Version   : 9.5(2)
        Serial No.: FCH183770GD
        CCL IP    : 127.2.1.2
        CCL MAC   : 0015.c500.019f
        Last join : 01:18:34 UTC Nov 4 2015
        Last leave: N/A
Other members in the cluster:
    Unit "unit-1-3" in state SLAVE
        ID        : 4
        Version   : 9.5(2)
        Serial No.: FCH19057ML0
        CCL IP    : 127.2.1.3
        CCL MAC   : 0015.c500.018f
        Last join : 20:29:57 UTC Nov 4 2015
        Last leave: 20:24:55 UTC Nov 4 2015
    Unit "unit-1-1" in state SLAVE
        ID        : 1
        Version   : 9.5(2)
        Serial No.: FCH19057ML0
        CCL IP    : 127.2.1.1
        CCL MAC   : 0015.c500.017f
        Last join : 20:20:53 UTC Nov 4 2015
        Last leave: 20:18:15 UTC Nov 4 2015
```

- **show cluster info transport** {**asp** | **cp**}

  Shows transport related statistics for the following:

  - **asp** —Data plane transport statistics.

  - **cp** —Control plane transport statistics.

- **show cluster history**

Shows the cluster history.

Capturing Packets Cluster-Wide

See the following command for capturing packets in a cluster:

cluster exec capture

To support cluster-wide troubleshooting, you can enable capture of cluster-specific traffic on the control unit using the **cluster exec capture** command, which is then automatically enabled on all of the data units in the cluster.

Monitoring Cluster Resources

See the following command for monitoring cluster resources:

show cluster {cpu | memory | resource} [*options*], **show cluster chassis [cpu | memory | resource usage]**

Displays aggregated data for the entire cluster. The options available depends on the data type.

Monitoring Cluster Traffic

See the following command for monitoring cluster traffic:

- **show conn [detail | count], cluster exec show conn**

  The **show conn** command shows whether a flow is a director, backup, or forwarder flow. Use the **cluster exec show conn** command on any unit to view all connections. This command can show how traffic for a single flow arrives at different ASAs in the cluster. The throughput of the cluster is dependent on the efficiency and configuration of load balancing. This command provides an easy way to view how traffic for a connection is flowing through the cluster, and can help you understand how a load balancer might affect the performance of a flow.

  The following is sample output for the **show conn detail** command:

```
ciscoasa/ASA2/slave# show conn detail
15 in use, 21 most used
Cluster:
        fwd connections: 0 in use, 0 most used
        dir connections: 0 in use, 0 most used
        centralized connections: 0 in use, 44 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - LISP triggered flow owner mobility
       M - SMTP data, m - SIP media, n - GUP
       N - inspected by Snort
       O - outbound data, o - offloaded,
       P - inside back connection,
       Q - Diameter, q - SQL*Net data,
       R - outside acknowledged FIN,
       R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
       V - VPN orphan, W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

Cluster units to ID mappings:
  ID 0: unit-2-1
  ID 1: unit-1-1
  ID 2: unit-1-2
  ID 3: unit-2-2
  ID 4: unit-2-3
  ID 255: The default cluster member ID which indicates no ownership or affiliation
          with an existing cluster member
```

- **show cluster info [conn-distribution | packet-distribution | loadbalance]**

  The **show cluster info conn-distribution** and **show cluster info packet-distribution** commands show traffic distribution across all cluster units. These commands can help you to evaluate and adjust the external load balancer.

  The **show cluster info loadbalance** command shows connection rebalance statistics.

- **show cluster {access-list | conn [count] | traffic | user-identity | xlate}** [*options*], **show cluster chassis {access-list | conn | traffic | user-identity | xlate count}**

  Displays aggregated data for the entire cluster. The options available depends on the data type.

  See the following output for the **show cluster access-list** command:

```
ciscoasa# show cluster access-list
hitcnt display order: cluster-wide aggregated result, unit-A, unit-B,  unit-C, unit-D
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval
 300
access-list 101; 122 elements; name hash: 0xe7d586b5
access-list 101 line 1 extended permit tcp 192.168.143.0 255.255.255.0 any eq www
(hitcnt=0, 0, 0, 0, 0) 0x207a2b7d
access-list 101 line 2 extended permit tcp any 192.168.143.0 255.255.255.0 (hitcnt=0,
0, 0, 0, 0) 0xfe4f4947
access-list  101 line 3 extended permit tcp host 192.168.1.183 host 192.168.43.238
(hitcnt=1, 0, 0, 0, 1) 0x7b521307
access-list 101 line 4 extended permit tcp host 192.168.1.116 host 192.168.43.238
(hitcnt=0, 0, 0, 0, 0) 0x5795c069
access-list 101 line 5 extended permit tcp host 192.168.1.177 host 192.168.43.238
(hitcnt=1, 0, 0, 1, 0) 0x51bde7ee
access list 101 line 6 extended permit tcp host 192.168.1.177 host 192.168.43.13
(hitcnt=0, 0, 0, 0, 0) 0x1e68697c
access-list 101 line 7 extended permit tcp host 192.168.1.177 host 192.168.43.132
(hitcnt=2, 0, 0, 1, 1) 0xc1ce5c49
access-list 101 line 8 extended permit tcp host 192.168.1.177 host 192.168.43.192
(hitcnt=3, 0, 1, 1, 1) 0xb6f59512
access-list 101 line 9 extended permit tcp host 192.168.1.177 host 192.168.43.44
(hitcnt=0, 0, 0, 0, 0) 0xdc104200
access-list 101 line 10 extended permit tcp host 192.168.1.112 host 192.168.43.44
(hitcnt=429, 109, 107, 109, 104)
0xce4f281d
access-list 101 line 11 extended permit tcp host 192.168.1.170 host 192.168.43.238
(hitcnt=3, 1, 0, 0, 2) 0x4143a818
access-list 101 line 12 extended permit tcp host 192.168.1.170 host 192.168.43.169
(hitcnt=2, 0, 1, 0, 1) 0xb18dfea4
access-list 101 line 13 extended permit tcp host 192.168.1.170 host 192.168.43.229
```

```
(hitcnt=1, 1, 0, 0, 0) 0x21557d71
access-list 101 line 14 extended permit tcp host 192.168.1.170 host 192.168.43.106
(hitcnt=0, 0, 0, 0, 0) 0x7316e016
access-list 101 line 15 extended permit tcp host 192.168.1.170 host 192.168.43.196
(hitcnt=0, 0, 0, 0, 0) 0x013fd5b8
access-list 101 line 16 extended permit tcp host 192.168.1.170 host 192.168.43.75
(hitcnt=0, 0, 0, 0, 0) 0x2c7dba0d
```

To display the aggregated count of in-use connections for all units, enter:

```
ciscoasa# show cluster conn count
Usage Summary In Cluster:**********************************************
124 in use, fwd connection 0 in use, dir connection 0 in use, centralized connection
0 in use (Cluster-wide aggregated)

unit-1-1(LOCAL):*******************************************************
40 in use, 48 most used, fwd connection 0 in use, 0 most used, dir connection 0 in use,
0 most used, centralized connection 0 in use, 46 most used

unit-2-2:**************************************************************
18 in use, 40 most used, fwd connection 0 in use, 0 most used, dir connection 0 in use,
0 most used, centralized connection 0 in use, 45 most used
```

- **show asp cluster counter**

  This command is useful for datapath troubleshooting.

Monitoring Cluster Routing

See the following commands for cluster routing:

- **show route cluster**
- **debug route cluster**

  Shows cluster information for routing.

Configuring Logging for Clustering

See the following command for configuring logging for clustering:

logging device-id

Each unit in the cluster generates syslog messages independently. You can use the **logging device-id** command to generate syslog messages with identical or different device IDs to make messages appear to come from the same or different units in the cluster.

Debugging Clustering

See the following commands for debugging clustering:

- **debug cluster [ccp | datapath | fsm | general | hc | license | rpc | service-module | transport]**

  Shows debug messages for clustering.

- **debug service-module**

  Shows debug messages for blade level issues including health check issues between the supervisor and the application.

- **show cluster info trace**

  The **show cluster info trace** command shows the debug information for further troubleshooting.

  See the following output for the **show cluster info trace** command:

```
ciscoasa# show cluster info trace
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Send CCP message to all: CCP_MSG_KEEPALIVE from 80-1 at
MASTER
```

Reference for Clustering

This section includes more information about how clustering operates.

ASA Features and Clustering

Some ASA features are not supported with ASA clustering, and some are only supported on the control unit. Other features might have caveats for proper usage.

Unsupported Features with Clustering

These features cannot be configured with clustering enabled, and the commands will be rejected.

- Unified Communication features that rely on TLS Proxy
- The following application inspections:
  - CTIQBE
  - GTP
  - H323, H225, and RAS
  - IPsec passthrough
  - MGCP
  - MMP
  - RTSP
  - SCCP (Skinny)
  - WAAS
  - WCCP
- Botnet Traffic Filter

- Auto Update Server
- DHCP client, server, and proxy. DHCP relay is supported.
- Failover
- Dead Connection Detection (DCD)
- FIPS mode

Centralized Features for Clustering

The following features are only supported on the control unit, and are not scaled for the cluster.

Note Traffic for centralized features is forwarded from member units to the control unit over the cluster control link.

If you use the rebalancing feature, traffic for centralized features may be rebalanced to non-control units before the traffic is classified as a centralized feature; if this occurs, the traffic is then sent back to the control unit.

For centralized features, if the control unit fails, all connections are dropped, and you have to re-establish the connections on the new control unit.

- The following application inspections:
  - DCERPC
  - NetBIOS
  - PPTP
  - RADIUS
  - RSH
  - SUNRPC
  - TFTP
  - XDMCP
- Dynamic routing
- Static route tracking
- IGMP multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- PIM multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- Authentication and Authorization for network access. Accounting is decentralized.
- Filtering Services

Features Applied to Individual Units

These features are applied to each ASA unit, instead of the cluster as a whole or to the control unit.

- QoS—The QoS policy is synced across the cluster as part of configuration replication. However, the policy is enforced on each unit independently. For example, if you configure policing on output, then the conform rate and conform burst values are enforced on traffic exiting a particular ASA. In a cluster with 3 units and with traffic evenly distributed, the conform rate actually becomes 3 times the rate for the cluster.
- Threat detection—Threat detection works on each unit independently; for example, the top statistics is unit-specific. Port scanning detection, for example, does not work because scanning traffic will be load-balanced between all units, and one unit will not see all traffic.
- Resource management—Resource management in multiple context mode is enforced separately on each unit based on local usage.

AAA for Network Access and Clustering

AAA for network access consists of three components: authentication, authorization, and accounting. Authentication and authorization are implemented as centralized features on the clustering control unit with replication of the data structures to the cluster data units. If a control unit is elected, the new control unit will have all the information it needs to continue uninterrupted operation of the established authenticated users and their associated authorizations. Idle and absolute timeouts for user authentications are preserved when a control unit change occurs.

Accounting is implemented as a distributed feature in a cluster. Accounting is done on a per-flow basis, so the cluster unit owning a flow will send accounting start and stop messages to the AAA server when accounting is configured for a flow.

Connection Settings

Connection limits are enforced cluster-wide (see the **set connection conn-max**, **set connection embryonic-conn-max**, **set connection per-client-embryonic-max**, and **set connection per-client-max** commands). Each unit has an estimate of the cluster-wide counter values based on broadcast messages. Due to efficiency considerations, the configured connection limit across the cluster might not be enforced exactly at the limit number. Each unit may overestimate or underestimate the cluster-wide counter value at any given time. However, the information will get updated over time in a load-balanced cluster.

FTP and Clustering

- If FTP data channel and control channel flows are owned by different cluster members, then the data channel owner will periodically send idle timeout updates to the control channel owner and update the idle timeout value. However, if the control flow owner is reloaded, and the control flow is re-hosted, the parent/child flow relationship will not longer be maintained; the control flow idle timeout will not be updated.
- If you use AAA for FTP access, then the control channel flow is centralized on the control unit.

Identity Firewall and Clustering

Only the control unit retrieves the user-group from the AD and the user-ip mapping from the AD agent. The control unit then populates the user information to data units, and data units can make a match decision for user identity based on the security policy.

Multicast Routing and Clustering

The control unit handles all multicast routing packets and data packets until fast-path forwarding is established. After the connection is established, each data unit can forward multicast data packets.

NAT and Clustering

NAT can affect the overall throughput of the cluster. Inbound and outbound NAT packets can be sent to different ASAs in the cluster, because the load balancing algorithm relies on IP addresses and ports, and NAT causes inbound and outbound packets to have different IP addresses and/or ports. When a packet arrives at the ASA that is not the NAT owner, it is forwarded over the cluster control link to the owner, causing large amounts of traffic on the cluster control link. Note that the receiving unit does not create a forwarding flow to the owner, because the NAT owner may not end up creating a connection for the packet depending on the results of security and policy checks.

If you still want to use NAT in clustering, then consider the following guidelines:

- NAT pool address distribution for dynamic PAT—The control unit evenly pre-distributes addresses across the cluster. If a member receives a connection and they have no addresses assigned, then the connection is forwarded to the control unit for PAT. If a cluster member leaves the cluster (due to failure), a backup member will get the PAT IP address, and if the backup exhausts its normal PAT IP address, it can make use of the new address. Make sure to include at least as many NAT addresses as there are units in the cluster, plus at least one extra address, to ensure that each unit receives an address, and that a failed unit can get a new address if its old address is in use by the member that took over the address. Use the **show nat pool cluster** command to see the address allocations.

- Reusing a PAT pool in multiple rules—To use the same PAT pool in multiple rules, you must be careful about the interface selection in the rules. You must either use specific interfaces in all rules, or "any" in all rules. You cannot mix specific interfaces and "any" across the rules, or the system might not be able to match return traffic to the right node in the cluster. Using unique PAT pools per rule is the most reliable option.

- No round-robin—Round-robin for a PAT pool is not supported with clustering.

- Dynamic NAT xlates managed by the control unit—The control unit maintains and replicates the xlate table to data units. When a data unit receives a connection that requires dynamic NAT, and the xlate is not in the table, it requests the xlate from the control unit. The data unit owns the connection.

- Stale xlates—The xlate idle time on the connection owner does not get updated. Thus, the idle time might exceed the idle timeout. An idle timer value higher than the configured timeout with a refcnt of 0 is an indication of a stale xlate.

- Per-session PAT feature—Although not exclusive to clustering, the per-session PAT feature improves the scalability of PAT and, for clustering, allows each data unit to own PAT connections; by contrast, multi-session PAT connections have to be forwarded to and owned by the control unit. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate, whereas ICMP and all other UDP traffic uses multi-session. You can configure per-session NAT rules to change these defaults for TCP and UDP, but you cannot configure per-session PAT for ICMP. For traffic that benefits from multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT for the associated TCP ports (the UDP ports for those H.323 and SIP are already multi-session by default). For more information about per-session PAT, see the firewall configuration guide.

- No static PAT for the following inspections—
  - FTP

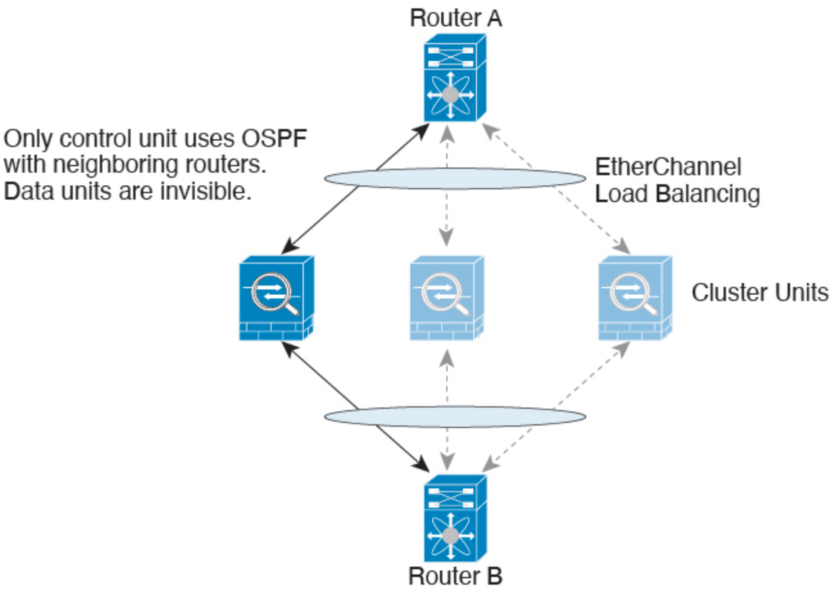
- PPTP
- RSH
- SQLNET
- TFTP
- XDMCP
- SIP

- If you have an extremely large number of NAT rules, over ten thousand, you should enable the transactional commit model using the **asp rule-engine transactional-commit nat** command in the device CLI. Otherwise, the unit might not be able to join the cluster.

Dynamic Routing and Clustering

The routing process only runs on the control unit, and routes are learned through the control unit and replicated to secondaries. If a routing packet arrives at a data unit, it is redirected to the control unit.

Figure 48: Dynamic Routing


Router A
Only control unit uses OSPF with neighboring routers. Data units are invisible.
EtherChannel Load Balancing
Cluster Units
Router B

After the data units learn the routes from the control unit, each unit makes forwarding decisions independently.

The OSPF LSA database is not synchronized from the control unit to data units. If there is a control unit switchover, the neighboring router will detect a restart; the switchover is not transparent. The OSPF process picks an IP address as its router ID. Although not required, you can assign a static router ID to ensure a consistent router ID is used across the cluster. See the OSPF Non-Stop Forwarding feature to address the interruption.

SIP Inspection and Clustering

A control flow can be created on any unit (due to load balancing); its child data flows must reside on the same unit.

TLS Proxy configuration is not supported.

SNMP and Clustering

An SNMP agent polls each individual ASA by its Local IP address. You cannot poll consolidated data for the cluster.

You should always use the Local address, and not the Main cluster IP address for SNMP polling. If the SNMP agent polls the Main cluster IP address, if a new control unit is elected, the poll to the new control unit will fail.

When using SNMPv3 with clustering, if you add a new cluster unit after the initial cluster formation, then SNMPv3 users are not replicated to the new unit.You must re-add them on the control unit to force the users to replicate to the new unit, or directly on the data unit.

Syslog and NetFlow and Clustering

- Syslog—Each unit in the cluster generates its own syslog messages. You can configure logging so that each unit uses either the same or a different device ID in the syslog message header field. For example, the hostname configuration is replicated and shared by all units in the cluster. If you configure logging to use the hostname as the device ID, syslog messages generated by all units look as if they come from a single unit. If you configure logging to use the local-unit name that is assigned in the cluster bootstrap configuration as the device ID, syslog messages look as if they come from different units.
- NetFlow—Each unit in the cluster generates its own NetFlow stream. The NetFlow collector can only treat each ASA as a separate NetFlow exporter.

Cisco TrustSec and Clustering

Only the control unit learns security group tag (SGT) information. The control unit then populates the SGT to data units, and data units can make a match decision for SGT based on the security policy.

Performance Scaling Factor

When you combine multiple units into a cluster, you can expect the total cluster performance to be approximately:

- 80% of the combined TCP or CPS throughput
- 90% of the combined UDP throughput
- 60% of the combined Ethernet MIX (EMIX) throughput, depending on the traffic mix.

For example, for TCP throughput, the Firepower 9300 with 3 SM-44 modules can handle approximately 135 Gbps of real world firewall traffic when running alone. For 2 chassis, the maximum combined throughput will be approximately 80% of 270 Gbps (2 chassis x 135 Gbps): 216 Gbps.

Control Unit Election

Members of the cluster communicate over the cluster control link to elect a control unit as follows:

1. When you deploy the cluster, each unit broadcasts an election request every 3 seconds.

2. Any other units with a higher priority respond to the election request; the priority is set when you deploy the cluster and is not configurable.
3. If after 45 seconds, a unit does not receive a response from another unit with a higher priority, then it becomes the control unit.

Note If multiple units tie for the highest priority, the cluster unit name and then the serial number is used to determine the control unit.

4. If a unit later joins the cluster with a higher priority, it does not automatically become the control unit; the existing control unit always remains as the control unit unless it stops responding, at which point a new control unit is elected.
5. In a "split brain" scenario when there are temporarily multiple control units, then the unit with highest priority retains the role while the other units return to data unit roles.

Note You can manually force a unit to become the control unit. For centralized features, if you force a control unit change, then all connections are dropped, and you have to re-establish the connections on the new control unit.

High Availability Within the Cluster

Clustering provides high availability by monitoring chassis, unit, and interface health and by replicating connection states between units.

Chassis-Application Monitoring

Chassis-application health monitoring is always enabled. The Firepower 9300 chassis supervisor checks the ASA application periodically (every second). If the ASA is up and cannot communicate with the Firepower 9300 chassis supervisor for 3 seconds, the ASA generates a syslog message and leaves the cluster.

If the Firepower 9300 chassis supervisor cannot communicate with the application after 45 seconds, it reloads the ASA. If the ASA cannot communicate with the supervisor, it removes itself from the cluster.

Unit Health Monitoring

Each unit periodically sends a broadcast keepalivekeepalive packet over the cluster control link. If the control unit does not receive any keepalivekeepalive packets or other packets from a data unit within the configurable timeout period, then the control unit removes the data unit from the cluster. If the data units do not receive packets from the control unit, then a new control unit is elected from the remaining members.

If units cannot reach each other over the cluster control link because of a network failure and not because a unit has actually failed, then the cluster may go into a "split brain" scenario where isolated data units will elect their own control units. For example, if a router fails between two cluster locations, then the original control unit at location 1 will remove the location 2 data units from the cluster. Meanwhile, the units at location 2 will elect their own control unit and form their own cluster. Note that asymmetric traffic may fail in this scenario. After the cluster control link is restored, then the control unit that has the higher priority will keep the control unit’s role. See Control Unit Election, on page 391 for more information.

Interface Monitoring

Each unit monitors the link status of all hardware interfaces in use, and reports status changes to the control unit. When you enable health monitoring, all physical interfaces are monitored by default (including the main EtherChannel for EtherChannel interfaces). Only named interfaces that are in an Up state can be monitored. For example, all member ports of an EtherChannel must fail before a *named* EtherChannel is removed from the cluster (depending on your minimum port bundling setting). You can optionally disable monitoring per interface.

If a monitored interface fails on a particular unit, but it is active on other units, then the unit is removed from the cluster. The amount of time before the ASA removes a member from the cluster depends on whether the unit is an established member or is joining the cluster. The ASA does not monitor interfaces for the first 90 seconds that a unit joins the cluster. Interface status changes during this time will not cause the ASA to be removed from the cluster. For an established member, the unit is removed after 500 ms.

Status After Failure

When a unit in the cluster fails, the connections hosted by that unit are seamlessly transferred to other units; state information for traffic flows is shared over the control unit's cluster control link.

If the control unit fails, then another member of the cluster with the highest priority (lowest number) becomes the control unit.

The ASA automatically tries to rejoin the cluster, depending on the failure event.

Note When the ASA becomes inactive and fails to automatically rejoin the cluster, all data interfaces are shut down; only the management-only interface can send and receive traffic. The management interface remains up using the IP address the unit received from the cluster IP pool. However if you reload, and the unit is still inactive in the cluster, the management interface is disabled. You must use the console port for any further configuration.

Rejoining the Cluster

After a cluster member is removed from the cluster, how it can rejoin the cluster depends on why it was removed:

- Failed cluster control link—After you resolve the problem with the cluster control link, you must manually rejoin the cluster by re-enabling clustering at the ASA console port by entering **cluster group** *name*, and then **enable**.
- Failed data interface—The ASA automatically tries to rejoin at 5 minutes, then at 10 minutes, and finally at 20 minutes. If the join is not successful after 20 minutes, then the ASA disables clustering. After you resolve the problem with the data interface, you have to manually enable clustering at the ASA console port by entering **cluster group** *name*, and then **enable**.
- Failed unit—If the unit was removed from the cluster because of a unit health check failure, then rejoining the cluster depends on the source of the failure. For example, a temporary power failure means the unit will rejoin the cluster when it starts up again as long as the cluster control link is up. The unit attempts to rejoin the cluster every 5 seconds.
- Failed Chassis-Application Communication—When the ASA detects that the chassis-application health has recovered, the ASA tries to rejoin the cluster automatically.

- Internal error—Internal failures include: application sync timeout; inconsistent application statuses; and so on. After you resolve the problem, you must manually rejoin the cluster by re-enabling clustering at the ASA console port by entering **cluster group** *name*, and then **enable**.

Data Path Connection State Replication

Every connection has one owner and at least one backup owner in the cluster. The backup owner does not take over the connection in the event of a failure; instead, it stores TCP/UDP state information, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner is usually also the director.

Some traffic requires state information above the TCP or UDP layer. See the following table for clustering support or lack of support for this kind of traffic.

Table 15: Features Replicated Across the Cluster

Traffic	State Support	Notes
Up time	Yes	Keeps track of the system up time.
ARP Table	Yes	Transparent mode only.
MAC address table	Yes	Transparent mode only.
User Identity	Yes	Includes AAA rules (uauth) and identity firewall.
IPv6 Neighbor database	Yes	—
Dynamic routing	Yes	—
SNMP Engine ID	**No**	—
Centralized VPN (Site-to-Site)	**No**	VPN sessions will be disconnected if the control unit fails.

How the Cluster Manages Connections

Connections can be load-balanced to multiple members of the cluster. Connection roles determine how connections are handled in both normal operation and in a high availability situation.

Connection Roles

See the following roles defined for each connection:

- Owner—Usually, the unit that initially receives the connection. The owner maintains the TCP state and processes packets. A connection has only one owner. If the original owner fails, then when new units receive packets from the connection, the director chooses a new owner from those units.
- Backup owner—The unit that stores TCP/UDP state information received from the owner, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner does not take over the connection in the event of a failure. If the owner becomes unavailable, then the first

unit to receive packets from the connection (based on load balancing) contacts the backup owner for the relevant state information so it can become the new owner.

As long as the director (see below) is not the same unit as the owner, then the director is also the backup owner. If the owner chooses itself as the director, then a separate backup owner is chosen.

- Director—The unit that handles owner lookup requests from forwarders. When the owner receives a new connection, it chooses a director based on a hash of the source/destination IP address and ports, and sends a message to the director to register the new connection. If packets arrive at any unit other than the owner, the unit queries the director about which unit is the owner so it can forward the packets. A connection has only one director. If a director fails, the owner chooses a new director.

  As long as the director is not the same unit as the owner, then the director is also the backup owner (see above). If the owner chooses itself as the director, then a separate backup owner is chosen.

- Forwarder—A unit that forwards packets to the owner. If a forwarder receives a packet for a connection it does not own, it queries the director for the owner, and then establishes a flow to the owner for any other packets it receives for this connection. The director can also be a forwarder. Note that if a forwarder receives the SYN-ACK packet, it can derive the owner directly from a SYN cookie in the packet, so it does not need to query the director. (If you disable TCP sequence randomization, the SYN cookie is not used; a query to the director is required.) For short-lived flows such as DNS and ICMP, instead of querying, the forwarder immediately sends the packet to the director, which then sends them to the owner. A connection can have multiple forwarders; the most efficient throughput is achieved by a good load-balancing method where there are no forwarders and all packets of a connection are received by the owner.

  Note We do not recommend disabling TCP sequence randomization when using clustering. There is a small chance that some TCP sessions won't be established, because the SYN/ACK packet might be dropped.

- Fragment Owner—For fragmented packets, cluster units that receive a fragment determine a fragment owner using a hash of the fragment source IP address, destination IP address, and the packet ID. All fragments are then forwarded to the fragment owner over the cluster control link. Fragments may be load-balanced to different cluster units, because only the first fragment includes the 5-tuple used in the switch load balance hash. Other fragments do not contain the source and destination ports and may be load-balanced to other cluster units. The fragment owner temporarily reassembles the packet so it can determine the director based on a hash of the source/destination IP address and ports. If it is a new connection, the fragment owner will register to be the connection owner. If it is an existing connection, the fragment owner forwards all fragments to the provided connection owner over the cluster control link. The connection owner will then reassemble all fragments.

When a connection uses Port Address Translation (PAT), then the PAT type (per-session or multi-session) influences which member of the cluster becomes the owner of a new connection:

- Per-session PAT—The owner is the unit that receives the initial packet in the connection.

  By default, TCP and DNS UDP traffic use per-session PAT.

- Multi-session PAT—The owner is always the control unit. If a multi-session PAT connection is initially received by a data unit, then the data unit forwards the connection to the control unit.

  By default, UDP (except for DNS UDP) and ICMP traffic use multi-session PAT, so these connections are always owned by the control unit.

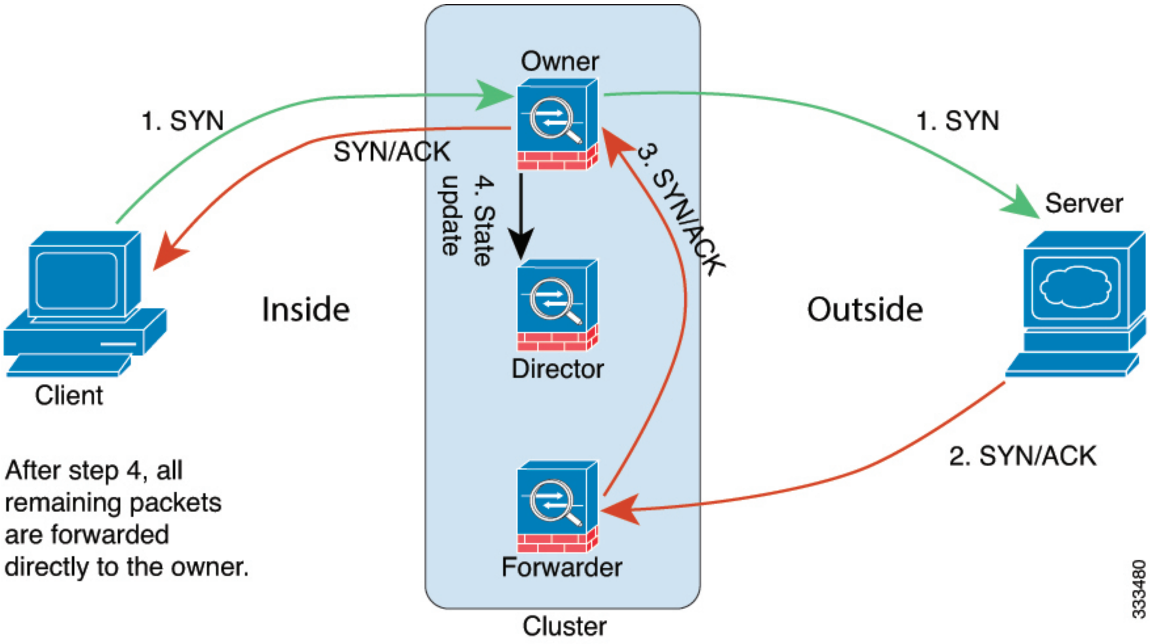
You can change the per-session PAT defaults for TCP and UDP so connections for these protocols are handled per-session or multi-session depending on the configuration. For ICMP, you cannot change from the default multi-session PAT. For more information about per-session PAT, see the firewall configuration guide.

New Connection Ownership

When a new connection is directed to a member of the cluster via load balancing, that unit owns both directions of the connection. If any connection packets arrive at a different unit, they are forwarded to the owner unit over the cluster control link. If a reverse flow arrives at a different unit, it is redirected back to the original unit.

Sample Data Flow

The following example shows the establishment of a new connection.



1. The SYN packet originates from the client and is delivered to one ASA (based on the load balancing method), which becomes the owner. The owner creates a flow, encodes owner information into a SYN cookie, and forwards the packet to the server.
2. The SYN-ACK packet originates from the server and is delivered to a different ASA (based on the load balancing method). This ASA is the forwarder.
3. Because the forwarder does not own the connection, it decodes owner information from the SYN cookie, creates a forwarding flow to the owner, and forwards the SYN-ACK to the owner.
4. The owner sends a state update to the director, and forwards the SYN-ACK to the client.
5. The director receives the state update from the owner, creates a flow to the owner, and records the TCP state information as well as the owner. The director acts as the backup owner for the connection.
6. Any subsequent packets delivered to the forwarder will be forwarded to the owner.

7. If packets are delivered to any additional units, it will query the director for the owner and establish a flow.

8. Any state change for the flow results in a state update from the owner to the director.

History for ASA Clustering on the Firepower 4100/9300

Feature Name	Version	Feature Information
Intra-chassis ASA Clustering for the Firepower 9300	9.4(1.150)	You can cluster up to 3 security modules within the Firepower 9300 chassis. All modules in the chassis must belong to the cluster. We introduced the following commands: **cluster replication delay, debug service-module, management-only individual, show cluster chassis**

PART III

Interfaces

CHAPTER 11

Basic Interface Configuration

This chapter includes basic interface configuration including Ethernet settings and Jumbo frame configuration.

Note For multiple context mode, complete all tasks in this section in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Note For the ASA Services Module interfaces, see the ASA Services Module quick start guide.

For the Firepower 9300 chassis, you configure basic interface settings in the FXOS operating system. See the configuration or getting started guide for your chassis for more information.

About Basic Interface Configuration

This section describes interface features and special interfaces.

Auto-MDI/MDIX Feature

For RJ-45 interfaces, the default auto-negotiation setting also includes the Auto-MDI/MDIX feature. Auto-MDI/MDIX eliminates the need for crossover cabling by performing an internal crossover when a straight cable is detected during the auto-negotiation phase. Either the speed or duplex must be set to auto-negotiate to enable Auto-MDI/MDIX for the interface. If you explicitly set both the speed and duplex to a fixed value, thus disabling auto-negotiation for both settings, then Auto-MDI/MDIX is also disabled. For

Gigabit Ethernet, when the speed and duplex are set to 1000 and full, then the interface always auto-negotiates; therefore Auto-MDI/MDIX is always enabled and you cannot disable it.

Management Interface

The management interface, depending on your model, is a separate interface just for management traffic.

Management Interface Overview

You can manage the ASA by connecting to:

- Any through-traffic interface
- A dedicated Management *Slot*/*Port* interface (if available for your model)

You may need to configure management access to the interface according to Management Access, on page 837.

Management *Slot*/*Port* Interface

The following table shows the Management interfaces per model.

Table 16: Management Interfaces Per Model

Model	Management 0/0	Management 0/1	Management 1/0	Management 1/1	Configurable for Through Traffic	Subinterfaces Allowed
Firepower 9300	N/A The interface ID depends on the physical mgmt-type interface that you assigned to the ASA logical device	—	—	—	—	**Yes**
ASA 5506-X	—	—	—	**Yes**	—	—
ASA 5508-X	—	—	—	**Yes**	—	—
ASA 5512-X	**Yes**	—	—	—	—	—
ASA 5515-X	**Yes**	—	—	—	—	—
ASA 5516-X	—	—	—	**Yes**	—	—
ASA 5525-X	**Yes**	—	—	—	—	—
ASA 5545-X	**Yes**	—	—	—	—	—
ASA 5555-X	**Yes**	—	—	—	—	—

Model	Management 0/0	Management 0/1	Management 1/0	Management 1/1	Configurable for Through Traffic	Subinterfaces Allowed
ASA 5585-X	**Yes**	**Yes**	**Yes** If you installed an SSP in slot 1, then Management 1/0 and 1/1 provide management access to the SSP in slot 1 only.	**Yes**	**Yes**	**Yes**
ISA 3000	—	—	—	**Yes**	—	—
ASASM	—	—	—	—	—	—
ASAv	**Yes**	—	—	—	—	—

Note If you installed a module, then the module management interface(s) provides management access for the module only. For models with software modules, the software module uses the same physical Management interface as the ASA.

Use Any Interface for Management-Only Traffic

You can use any interface as a dedicated management-only interface by configuring it for management traffic, including an EtherChannel interface (see the **management-only** command).

Management Interface for Transparent Mode

In transparent firewall mode, in addition to the maximum allowed through-traffic interfaces, you can also use the Management interface (either the physical interface, a subinterface (if supported for your model), or an EtherChannel interface comprised of Management interfaces (ASA 5585-X only)) as a separate management-only interface. You cannot use any other interface types as Management interfaces. For the Firepower 9300 chassis, the management interface ID depends on the mgmt-type interface that you assigned to the ASA logical device.

In multiple context mode, you cannot share any interfaces, including the Management interface, across contexts. To provide management per context on Firepower models and the ASA 5585-X, you can create subinterfaces of the Management interface and allocate a Management subinterface to each context. However, ASA models other than the ASA 5585-X do not allow subinterfaces on the Management interface, so per-context management for these models requires you to connect to a data interface. For the Firepower 9300 chassis, the management interface and its subinterfaces are not recognized as specially-allowed management interfaces within the contexts; you must treat a management subinterface as a data interface in this case and add it to a BVI.

The management interface is not part of a normal bridge group. Note that for operational purposes, it is part of a non-configurable bridge group.

Note In transparent firewall mode, the management interface updates the MAC address table in the same manner as a data interface; therefore you should not connect both a management and a data interface to the same switch unless you configure one of the switch ports as a routed port (by default Catalyst switches share a MAC address for all VLAN switch ports). Otherwise, if traffic arrives on the management interface from the physically-connected switch, then the ASA updates the MAC address table to use the *management* interface to access the switch, instead of the data interface. This action causes a temporary traffic interruption; the ASA will not re-update the MAC address table for packets from the switch to the data interface for at least 30 seconds for security reasons.

No Support for Redundant Management Interfaces

Redundant interfaces do not support Management *slot*/*port* interfaces as members. You can, however, set a redundant interface comprised of non-Management interfaces as management-only.

Management Interface Characteristics for ASA Models

The Management interface for ASA 5500-X models except for the ASA 5585-X has the following characteristics:

- No through traffic support
- No subinterface support
- No priority queue support
- No multicast MAC support
- The software module shares the Management interface. Separate MAC addresses and IP addresses are supported for the ASA and module. You must perform configuration of the module IP address within the module operating system. However, physical characteristics (such as enabling the interface) are configured on the ASA.

Licensing for Basic Interface Configuration

Model	License Requirement
ASA 5585-X	Interface Speed for SSP-10 and SSP-20: • Base License—1-Gigabit Ethernet for fiber interfaces • 10 GE I/O License (Security Plus)—10-Gigabit Ethernet for fiber interfaces • (SSP-40 and SSP-60 support 10-Gigabit Ethernet by default.)

Guidelines for Basic Interface Configuration

Transparent Firewall Mode

For multiple context, transparent mode, each context must use different interfaces; you cannot share an interface across contexts.

Failover

You cannot share a failover or state interface with a data interface.

Additional Guidelines

Some management-related services are not available until a non-management interface is enabled, and the the ASA achieves a “System Ready” state. The ASA generates the following syslog message when it is in a “System Ready” state:

```
%ASA-6-199002: Startup completed.  Beginning operation.
```

Default Settings for Basic Interface Configuration

This section lists default settings for interfaces if you do not have a factory default configuration.

Default State of Interfaces

The default state of an interface depends on the type and the context mode.

In multiple context mode, all allocated interfaces are enabled by default, no matter what the state of the interface is in the system execution space. However, for traffic to pass through the interface, the interface also has to be enabled in the system execution space. If you shut down an interface in the system execution space, then that interface is down in all contexts that share it.

In single mode or in the system execution space, interfaces have the following default states:

- Physical interfaces—Disabled.
- Redundant Interfaces—Enabled. However, for traffic to pass through the redundant interface, the member physical interfaces must also be enabled.
- VLAN subinterfaces—Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.
- VXLAN VNI interfaces—Enabled.
- EtherChannel port-channel interfaces (ASA models)—Enabled. However, for traffic to pass through the EtherChannel, the channel group physical interfaces must also be enabled.
- EtherChannel port-channel interfaces (Firepower models)—Disabled.

Note For the Firepower 9300, you can administratively enable and disable interfaces in both the chassis and on the ASA. For an interface to be operational, the interface must be enabled in both operating systems. Because the interface state is controlled independently, you may have a mismatch between the chassis and the ASA.

Default Speed and Duplex

- By default, the speed and duplex for copper (RJ-45) interfaces are set to auto-negotiate.
- For fiber interfaces for the 5585-X, the speed is set for automatic link negotiation.

Default Connector Type

Some models include two connector types: copper RJ-45 and fiber SFP. RJ-45 is the default. You can configure the ASA to use the fiber SFP connectors.

Default MAC Addresses

By default, the physical interface uses the burned-in MAC address, and all subinterfaces of a physical interface use the same burned-in MAC address.

Enable the Physical Interface and Configure Ethernet Parameters

This section describes how to:

- Enable the physical interface
- Set a specific speed and duplex (if available)
- Enable pause frames for flow control

Before you begin

For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Specify the interface you want to configure:

interface *physical_interface*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

The *physical_interface* ID includes the type, slot, and port number as type[slot/]port.

The physical interface types include the following:

- **gigabitethernet**
- **tengigabitethernet**
- **management**

Enter the type followed by *slot*/*port*, for example, **gigabitethernet0/1**. A space is optional between the type and the slot/port.

Step 2 (Optional) Set the media type to SFP, if available for your model:

media-type sfp

To restore the default RJ-45, enter the **media-type rj45** command.

Step 3 (Optional) Set the speed:

speed {**auto** | **10** | **100** | **1000** | **10000** | **nonegotiate**}

Example:

```
ciscoasa(config-if)# speed 100
```

For RJ-45 interfaces, the default setting is **auto**.

For SFP interfaces, the default setting is **no speed nonegotiate**, which sets the speed to the maximum speed (up to 1000 Mbps) and enables link negotiation for flow-control parameters and remote fault information. For 10GB interfaces, this option sets the speed down to 1000 Mbps. The **nonegotiate** keyword is the only keyword available for SFP interfaces. The **speed nonegotiate** command disables link negotiation.

Step 4 (Optional) Set the duplex for RJ-45 interfaces:

duplex {**auto** | **full** | **half**}

Example:

```
ciscoasa(config-if)# duplex full
```

The **auto** setting is the default. The duplex setting for an EtherChannel interface must be **full** or **auto**.

Step 5 (Optional) Enable pause (XOFF) frames for flow control on GigabitEthernet and TenGigabitEthernet interfaces:

flowcontrol send on [*low_water high_water pause_time*] [**noconfirm**]

Example:

```
ciscoasa(config-if)# flowcontrol send on 95 200 10000
```

If you have a traffic burst, dropped packets can occur if the burst exceeds the buffering capacity of the FIFO buffer on the NIC and the receive ring buffers. Enabling pause frames for flow control can alleviate this issue. Pause (XOFF) and XON frames are generated automatically by the NIC hardware based on the FIFO buffer usage. A pause frame is sent when the buffer usage exceeds the high-water mark. The default *high_water* value is 128 KB (10 GigabitEthernet) and 24 KB (1 GigabitEthernet); you can set it between 0 and 511 (10 GigabitEthernet) or 0 and 47 KB (1 GigabitEthernet). After a pause is sent, an XON frame can be sent when the buffer usage is reduced below the low-water mark. By default, the *low_water* value is 64 KB (10

GigabitEthernet) and 16 KB (1 GigabitEthernet); you can set it between 0 and 511 (10 GigabitEthernet) or 0 and 47 KB (1 GigabitEthernet). The link partner can resume traffic after receiving an XON, or after the XOFF expires, as controlled by the timer value in the pause frame. The default *pause_time* value is 26624; you can set it between 0 and 65535. If the buffer usage is consistently above the high-water mark, pause frames are sent repeatedly, controlled by the pause refresh threshold value.

When you use this command, you see the following warning:

```
Changing flow-control parameters will reset the interface. Packets may be lost during the
reset.
Proceed with flow-control changes?
```

To change the parameters without being prompted, use the **noconfirm** keyword.

Note Only flow control frames defined in 802.3x are supported. Priority-based flow control is not supported.

Step 6 Enable the interface:

no shutdown

Example:

```
ciscoasa(config-if)# no shutdown
```

To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.

Enable Jumbo Frame Support (ASA Models)

A jumbo frame is an Ethernet packet larger than the standard maximum of 1518 bytes (including Layer 2 header and VLAN header), up to 9216 bytes. You can enable support for jumbo frames for all interfaces by increasing the amount of memory to process Ethernet frames. Assigning more memory for jumbo frames might limit the maximum use of other features, such as ACLs. Note that the ASA MTU sets the payload size not including the Layer 2 (14 bytes) and VLAN header (4 bytes), so the maximum MTU is 9198, depending on your model.

Note This procedure only applies to ASA hardware models. Firepower models support jumbo frames by default.

Before you begin

- In multiple context mode, set this option in the system execution space.
- Changes in this setting require you to reload the ASA.

- Be sure to set the MTU for each interface that needs to transmit jumbo frames to a higher value than the default 1500; for example, set the value to 9198 using the **mtu** command. In multiple context mode, set the MTU within each context.
- Be sure to adjust the TCP MSS, either to disable it for non-IPsec traffic (use the **sysopt connection tcpmss 0** command), or to increase it in accord with the MTU.

Procedure

Enable jumbo frame support:

jumbo-frame reservation

Examples

The following example enables jumbo frame reservation, saves the configuration, and reloads the ASA:

```
ciscoasa(config)# jumbo-frame reservation
WARNING: this command will take effect after the running-config is saved
and the system has been rebooted. Command accepted.

ciscoasa(config)# write memory
Building configuration...
Cryptochecksum: 718e3706 4edb11ea 69af58d0 0a6b7cb5

70291 bytes copied in 3.710 secs (23430 bytes/sec)
[OK]
ciscoasa(config)# reload
Proceed with reload? [confirm] Y
```

Monitoring Interfaces

See the following commands.

Note For the Firepower 9300, some statistics are not shown using the ASA commands. You must view more detailed interface statistics using FXOS commands.

- /eth-uplink/fabric# **show interface**
- /eth-uplink/fabric# **show port-channel**
- /eth-uplink/fabric/interface# **show stats**

See the FXOS troubleshooting guide for more information.

- **show interface**

Displays interface statistics.

- **show interface ip brief**

  Displays interface IP addresses and status.

Examples for Basic Interfaces

See the following configuration examples.

Physical Interface Parameters Example

The following example configures parameters for the physical interface in single mode:

```
interface gigabitethernet 0/1
speed 1000
duplex full
no shutdown
```

Multiple Context Mode Example

The following example configures interface parameters in multiple context mode for the system configuration, and allocates the gigabitethernet 0/1.1 subinterface to contextA:

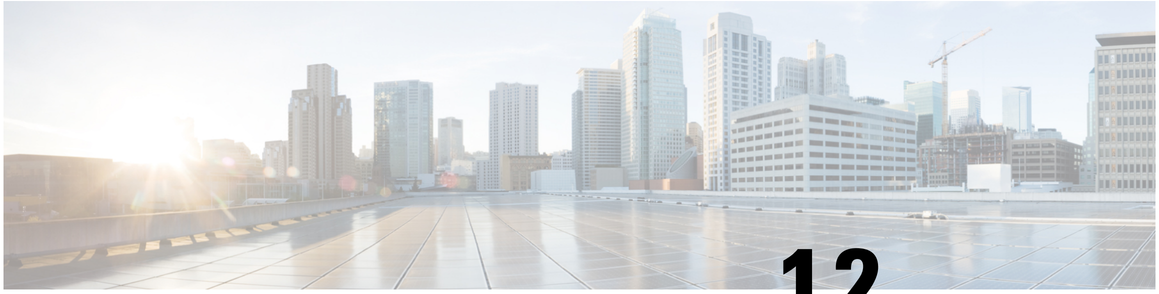
```
interface gigabitethernet 0/1
speed 1000
duplex full
no shutdown
interface gigabitethernet 0/1.1
vlan 101
context contextA
allocate-interface gigabitethernet 0/1.1
```

History for Basic Interface Configuration

Table 17: History for Interfaces

Feature Name	Releases	Feature Information
Support for Pause Frames for Flow Control on Gigabit Ethernet Interfaces	8.2(5)/8.4(2)	You can now enable pause (XOFF) frames for flow control for Gigabit Ethernet interfaces on all models. We modified the following command: **flowcontrol**.

Feature Name	Releases	Feature Information
Support for Pause Frames for Flow Control on the ASA 5580 Ten Gigabit Ethernet Interfaces	8.2(2)	You can now enable pause (XOFF) frames for flow control. This feature is also supported on the ASA 5585-X. We introduced the following command: **flowcontrol**.
Jumbo packet support for the ASA 5580	8.1(1)	The Cisco ASA 5580 supports jumbo frames. A jumbo frame is an Ethernet packet larger than the standard maximum of 1518 bytes (including Layer 2 header and FCS), up to 9216 bytes. You can enable support for jumbo frames for all interfaces by increasing the amount of memory to process Ethernet frames. Assigning more memory for jumbo frames might limit the maximum use of other features, such as ACLs. This feature is also supported on the ASA 5585-X. We introduced the following command: **jumbo-frame reservation**.
Gigabit Ethernet Support for the ASA 5510 Security Plus License	7.2(3)	The ASA 5510 now supports GE (Gigabit Ethernet) for port 0 and 1 with the Security Plus license. If you upgrade the license from Base to Security Plus, the capacity of the external Ethernet0/0 and Ethernet0/1 ports increases from the original FE (Fast Ethernet) (100 Mbps) to GE (1000 Mbps). The interface names will remain Ethernet 0/0 and Ethernet 0/1. Use the **speed** command to change the speed on the interface and use the **show interface** command to see what speed is currently configured for each interface.
Increased interfaces for the Base license on the ASA 5510	7.2(2)	For the Base license on the ASA 5510, the maximum number of interfaces was increased from 3 plus a management interface to unlimited interfaces.



CHAPTER 12

EtherChannel and Redundant Interfaces

This chapter tells how to configure EtherChannels and redundant interfaces.

Note For multiple context mode, complete all tasks in this section in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

For ASA cluster interfaces, which have special requirements, see ASA Cluster, on page 271.

Note For Firepower 9300 chassis, EtherChannel interfaces are configured in the FXOS operating system. Redundant interfaces are not supported. See the configuration or getting started guide for your chassis for more information.

About EtherChannels and Redundant Interfaces

This section describes EtherChannels and Redundant Interfaces.

About Redundant Interfaces (ASA Platform Only)

A logical redundant interface consists of a pair of physical interfaces: an active and a standby interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the ASA reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as device-level failover if desired.

You can configure up to 8 redundant interface pairs.

Redundant Interface MAC Address

The redundant interface uses the MAC address of the first physical interface that you add. If you change the order of the member interfaces in the configuration, then the MAC address changes to match the MAC address of the interface that is now listed first. Alternatively, you can assign a manual MAC address to the redundant interface, which is used regardless of the member interface MAC addresses. When the active interface fails over to the standby, the same MAC address is maintained so that traffic is not disrupted.

Related Topics

Configure the MTU and TCP MSS, on page 482

Configure Multiple Contexts, on page 185

About EtherChannels

An 802.3ad EtherChannel is a logical interface (called a port-channel interface) consisting of a bundle of individual Ethernet links (a channel group) so that you increase the bandwidth for a single network. A port channel interface is used in the same way as a physical interface when you configure interface-related features.

You can configure up to 48 EtherChannels, depending on how many interfaces your model supports.

Channel Group Interfaces

Each channel group can have up to 16 active interfaces. For switches that support only 8 active interfaces, you can assign up to 16 interfaces to a channel group: while only 8 interfaces can be active, the remaining interfaces can act as standby links in case of interface failure. For 16 active interfaces, be sure that your switch supports the feature (for example, the Cisco Nexus 7000 with F2-Series 10 Gigabit Ethernet Module).

All interfaces in the channel group must be the same type and speed. The first interface added to the channel group determines the correct type and speed.

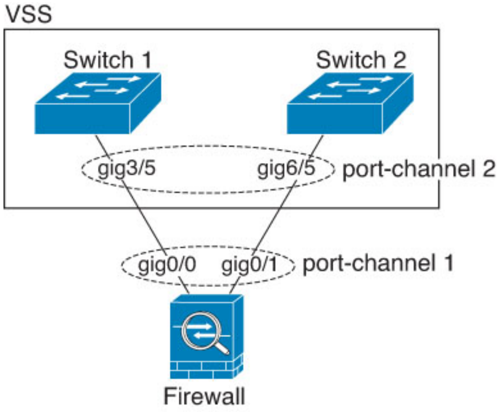
The EtherChannel aggregates the traffic across all the available active interfaces in the channel. The interface is selected using a proprietary hash algorithm, based on source or destination MAC addresses, IP addresses, TCP and UDP port numbers and VLAN numbers.

Connecting to an EtherChannel on Another Device

The device to which you connect the ASA EtherChannel must also support 802.3ad EtherChannels; for example, you can connect to the Catalyst 6500 switch or the Cisco Nexus 7000.

When the switch is part of a Virtual Switching System (VSS) or Virtual Port Channel (vPC), then you can connect ASA interfaces within the same EtherChannel to separate switches in the VSS/vPC. The switch interfaces are members of the same EtherChannel port-channel interface, because the separate switches act like a single switch.

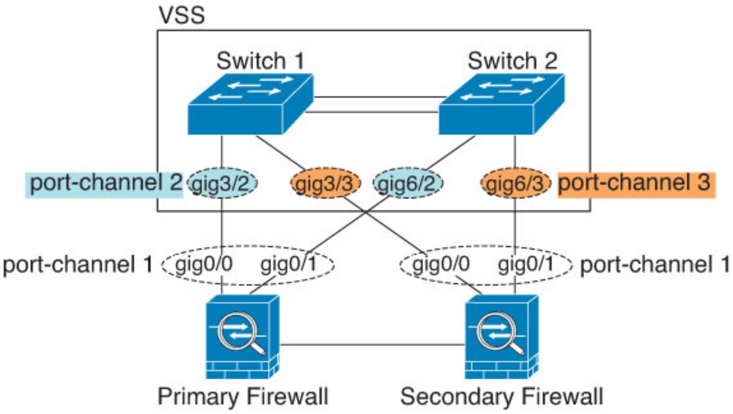
Figure 49: Connecting to a VSS/vPC



Note If the ASA is in transparent firewall mode, and you place the ASA between two sets of VSS/vPC switches, then be sure to disable Unidirectional Link Detection (UDLD) on any switch ports connected to the ASA with an EtherChannel. If you enable UDLD, then a switch port may receive UDLD packets sourced from both switches in the other VSS/vPC pair. The receiving switch will place the receiving interface in a down state with the reason "UDLD Neighbor mismatch".

If you use the ASA in an Active/Standby failover deployment, then you need to create separate EtherChannels on the switches in the VSS/vPC, one for each ASA. On each ASA, a single EtherChannel connects to both switches. Even if you could group all switch interfaces into a single EtherChannel connecting to both ASA (in this case, the EtherChannel will not be established because of the separate ASA system IDs), a single EtherChannel would not be desirable because you do not want traffic sent to the standby ASA.

Figure 50: Active/Standby Failover and VSS/vPC



Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) aggregates interfaces by exchanging the Link Aggregation Control Protocol Data Units (LACPDUs) between two network devices.

You can configure each physical interface in an EtherChannel to be:

- Active—Sends and receives LACP updates. An active EtherChannel can establish connectivity with either an active or a passive EtherChannel. You should use the active mode unless you need to minimize the amount of LACP traffic.
- Passive—Receives LACP updates. A passive EtherChannel can only establish connectivity with an active EtherChannel. Not supported on Firepower hardware models.
- On—The EtherChannel is always on, and LACP is not used. An "on" EtherChannel can only establish a connection with another "on" EtherChannel. Not supported on Firepower hardware models.

LACP coordinates the automatic addition and deletion of links to the EtherChannel without user intervention. It also handles misconfigurations and checks that both ends of member interfaces are connected to the correct channel group. "On" mode cannot use standby interfaces in the channel group when an interface goes down, and the connectivity and configurations are not checked.

Load Balancing

The ASA distributes packets to the interfaces in the EtherChannel by hashing the source and destination IP address of the packet (this criteria is configurable). The resulting hash is divided by the number of active links in a modulo operation where the resulting remainder determines which interface owns the flow. All packets with a *hash_value* **mod** *active_links* result of 0 go to the first interface in the EtherChannel, packets with a result of 1 go to the second interface, packets with a result of 2 go to the third interface, and so on. For example, if you have 15 active links, then the modulo operation provides values from 0 to 14. For 6 active links, the values are 0 to 5, and so on.

For a spanned EtherChannel in clustering, load balancing occurs on a per ASA basis. For example, if you have 32 active interfaces in the spanned EtherChannel across 8 ASAs, with 4 interfaces per ASA in the EtherChannel, then load balancing only occurs across the 4 interfaces on the ASA.

If an active interface goes down and is not replaced by a standby interface, then traffic is rebalanced between the remaining links. The failure is masked from both Spanning Tree at Layer 2 and the routing table at Layer 3, so the switchover is transparent to other network devices.

Related Topics

Customize the EtherChannel, on page 423

EtherChannel MAC Address

All interfaces that are part of the channel group share the same MAC address. This feature makes the EtherChannel transparent to network applications and users, because they only see the one logical connection; they have no knowledge of the individual links.

The port-channel interface uses the lowest numbered channel group interface MAC address as the port-channel MAC address. Alternatively you can manually configure a MAC address for the port-channel interface. In multiple context mode, you can automatically assign unique MAC addresses to *shared* interfaces, including an EtherChannel port interface. We recommend manually, or in multiple context mode for shared interfaces, automatically configuring a unique MAC address in case the group channel interface membership changes. If you remove the interface that was providing the port-channel MAC address, then the port-channel MAC address changes to the next lowest numbered interface, thus causing traffic disruption.

Guidelines for EtherChannels and Redundant Interfaces

Failover

- When you use a redundant or EtherChannel interface as a Failover link, it must be pre-configured on both units in the Failover pair; you cannot configure it on the primary unit and expect it to replicate to the secondary unit because *the Failover link itself is required for replication*.
- If you use a redundant or EtherChannel interface for the state link, no special configuration is required; the configuration can replicate from the primary unit as normal. For the Firepower 9300 chassis, all interfaces, including EtherChannels, need to be pre-configured on both units.
- You can monitor redundant or EtherChannel interfaces for Failover using the **monitor-interface** command; be sure to reference the logical redundant interface name. When an active member interface fails over to a standby interface, this activity does not cause the redundant or EtherChannel interface to appear to be failed when being monitored for device-level Failover. Only when all physical interfaces fail does the redundant or EtherChannel interface appear to be failed (for an EtherChannel interface, the number of member interfaces allowed to fail is configurable).
- If you use an EtherChannel interface for a Failover or state link, then to prevent out-of-order packets, only one interface in the EtherChannel is used. If that interface fails, then the next interface in the EtherChannel is used. You cannot alter the EtherChannel configuration while it is in use as a Failover link. To alter the configuration, you need to temporarily disable Failover, which prevents Failover from occurring for the duration.

Model Support

- You cannot add EtherChannels in ASA for the Firepower 4100/9300, ASASM, or the ASAv. The Firepower 4100/9300 supports EtherChannels, but you must perform all hardware configuration of EtherChannels in FXOS on the chassis.
- Redundant interfaces are only supported on the ASA 5500-X platform; they are not supported on the Firepower 4100/9300, ASASM, and ASAv.

Clustering

- When you use a redundant or EtherChannel interface as the Cluster Control Link, it must be pre-configured on all units in the cluster; you cannot configure it on the primary unit and expect it to replicate to member units because *the Cluster Control Link itself is required for replication*.
- To configure a spanned EtherChannel or an individual cluster interface, see the clustering chapter.

General Redundant Interface Guidelines

- You can configure up to 8 redundant interface pairs.
- All ASA configuration refers to the logical redundant interface instead of the member physical interfaces.
- You cannot use a redundant interface as part of an EtherChannel, nor can you use an EtherChannel as part of a redundant interface. You cannot use the same physical interfaces in a redundant interface and

an EtherChannel interface. You can, however, configure both types on the ASA if they do not use the same physical interfaces.

- If you shut down the active interface, then the standby interface becomes active.
- Redundant interfaces do not support Management *slot*/*port* interfaces as members. You can, however, set a redundant interface comprised of non-Management interfaces as management-only.

General EtherChannel Guidelines

- You can configure up to 48 EtherChannels, depending on how many interfaces are available on your model.
- Each channel group can have up to 16 active interfaces. For switches that support only 8 active interfaces, you can assign up to 16 interfaces to a channel group: while only 8 interfaces can be active, the remaining interfaces can act as standby links in case of interface failure. For 16 active interfaces, be sure that your switch supports the feature (for example, the Cisco Nexus 7000 with F2-Series 10 Gigabit Ethernet Module).
- All interfaces in the channel group must be the same media type and capacity, and must be set to the same speed and duplex. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface.
- The device to which you connect the ASA EtherChannel must also support 802.3ad EtherChannels.
- The ASA does not support LACPDUs that are VLAN-tagged. If you enable native VLAN tagging on the neighboring switch using the Cisco IOS **vlan dot1Q tag native** command, then the ASA will drop the tagged LACPDUs. Be sure to disable native VLAN tagging on the neighboring switch. In multiple context mode, these messages are not included in a packet capture, so that you cannot diagnose the issue easily.
- ASA 5500-X models do not support LACP rate fast; LACP always uses the normal rate. This setting is not configurable. Note that the Firepower 4100/9300, which configures EtherChannels in FXOS, has the LACP rate set to fast by default; on these platforms, the rate is configurable.
- In Cisco IOS software versions earlier than 15.1(1)S2, the ASA did not support connecting an EtherChannel to a switch stack. With default switch settings, if the ASA EtherChannel is connected cross stack, and if the primary switch is powered down, then the EtherChannel connected to the remaining switch will not come up. To improve compatibility, set the **stack-mac persistent timer** command to a large enough value to account for reload time; for example, 8 minutes or 0 for indefinite. Or, you can upgrade to more a more stable switch software version, such as 15.1(1)S2.
- All ASA configuration refers to the logical EtherChannel interface instead of the member physical interfaces.
- You cannot use a redundant interface as part of an EtherChannel, nor can you use an EtherChannel as part of a redundant interface. You cannot use the same physical interfaces in a redundant interface and an EtherChannel interface. You can, however, configure both types on the ASA if they do not use the same physical interfaces.

Default Settings for EtherChannels and Redundant Interfaces

This section lists default settings for interfaces if you do not have a factory default configuration.

Default State of Interfaces

The default state of an interface depends on the type and the context mode.

In multiple context mode, all allocated interfaces are enabled by default, no matter what the state of the interface is in the system execution space. However, for traffic to pass through the interface, the interface also has to be enabled in the system execution space. If you shut down an interface in the system execution space, then that interface is down in all contexts that share it.

In single mode or in the system execution space, interfaces have the following default states:

- Physical interfaces—Disabled.
- Redundant Interfaces—Enabled. However, for traffic to pass through the redundant interface, the member physical interfaces must also be enabled.
- EtherChannel port-channel interfaces—Enabled. However, for traffic to pass through the EtherChannel, the channel group physical interfaces must also be enabled.

Configure a Redundant Interface

A logical redundant interface consists of a pair of physical interfaces: an active and a standby interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the ASA reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as failover if desired.

This section describes how to configure redundant interfaces.

Configure a Redundant Interface

This section describes how to create a redundant interface. By default, redundant interfaces are enabled.

Before you begin

- You can configure up to 8 redundant interface pairs.
- Redundant interface delay values are configurable, but by default the ASA inherits the default delay values based on the physical type of its member interfaces.
- Both member interfaces must be of the same physical type. For example, both must be GigabitEthernet.
- You cannot add a physical interface to the redundant interface if you configured a name for it. You must first remove the name using the **no nameif** command.
- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Caution If you are using a physical interface already in your configuration, removing the name will clear any configuration that refers to the interface.

Procedure

Step 1 Add the logical redundant interface:

interface redundant *number*

Example:

```
ciscoasa(config)# interface redundant 1
```

The *number* argument is an integer between 1 and 8.

You need to add at least one member interface to the redundant interface before you can configure logical parameters for it such as a name.

Step 2 Add the first member interface to the redundant interface:

member-interface *physical_interface*

Example:

```
ciscoasa(config-if)# member-interface gigabitethernet 0/0
```

Redundant interfaces do not support Management *slot*/*port* interfaces as members.

After you add the interface, any configuration for it (such as an IP address) is removed.

Step 3 Add the second member interface to the redundant interface:

member-interface *physical_interface*

Example:

```
ciscoasa(config-if)# member-interface gigabitethernet 0/1
```

Make sure the second interface is the same physical type as the first interface.

To remove a member interface, enter the **no member-interface** *physical_interface* command. You cannot remove both member interfaces from the redundant interface; the redundant interface requires at least one member interface.

Examples

The following example creates two redundant interfaces:

```
ciscoasa(config)# interface redundant 1
ciscoasa(config-if)# member-interface gigabitethernet 0/0
ciscoasa(config-if)# member-interface gigabitethernet 0/1
ciscoasa(config-if)# interface redundant 2
ciscoasa(config-if)# member-interface gigabitethernet 0/2
ciscoasa(config-if)# member-interface gigabitethernet 0/3
```

Change the Active Interface

By default, the active interface is the first interface listed in the configuration, if it is available.

Procedure

Step 1 To view which interface is active, enter the following command :

show interface redundant *number* **detail | grep Member**

Example:

```
ciscoasa# show interface redundant1 detail | grep Member
      Members GigabitEthernet0/3(Active), GigabitEthernet0/2
```

Step 2 Change the active interface:

redundant-interface redundant *number* **active-member** *physical_interface*

The **redundant***number* argument is the redundant interface ID, such as **redundant1**.

The *physical_interface* is the member interface ID that you want to be active.

Configure an EtherChannel

This section describes how to create an EtherChannel port-channel interface, assign interfaces to the EtherChannel, and customize the EtherChannel.

Add Interfaces to the EtherChannel

This section describes how to create an EtherChannel port-channel interface and assign interfaces to the EtherChannel. By default, port-channel interfaces are enabled.

Before you begin

- You can configure up to 48 EtherChannels, depending on how many interfaces your model has.
- Each channel group can have up to 16 active interfaces. For switches that support only 8 active interfaces, you can assign up to 16 interfaces to a channel group: while only eight interfaces can be active, the remaining interfaces can act as standby links in case of interface failure.

- To configure a spanned EtherChannel for clustering, see the clustering chapter instead of this procedure.
- All interfaces in the channel group must be the same media type and capacity, and must be set to the same speed and duplex. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface.. Note that for interfaces that you can configure to use either the RJ-45 or SFP connector, you can include both RJ-45 and SFP interfaces in the same EtherChannel.
- You cannot add a physical interface to the channel group if you configured a name for it. You must first remove the name using the **no nameif** command.
- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Caution If you are using a physical interface already in your configuration, removing the name will clear any configuration that refers to the interface.

Procedure

Step 1 Specify the interface you want to add to the channel group:

interface *physical_interface*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

The *physical_interface* ID includes the type, slot, and port number as type[slot/]port. This first interface in the channel group determines the type and speed for all other interfaces in the group.

In transparent mode, if you create a channel group with multiple Management interfaces, then you can use this EtherChannel as the management-only interface.

Step 2 Assign this physical interface to an EtherChannel:

channel-group *channel_id* **mode** {**active** | **passive** | **on**}

Example:

```
ciscoasa(config-if)# channel-group 1 mode active
```

The *channel_id* is an integer between 1 and 48. If the port-channel interface for this channel ID does not yet exist in the configuration, one will be added:

interface port-channel *channel_id*

We recommend using **active** mode.

Step 3 (Optional) Set the priority for a physical interface in the channel group:

lacp port-priority *number*

Example:

```
ciscoasa(config-if)# lacp port-priority 12345
```

The priority *number* is an integer between 1 and 65535. The default is 32768. The higher the number, the lower the priority. The ASA uses this setting to decide which interfaces are active and which are standby if you assign more interfaces than can be used. If the port priority setting is the same for all interfaces, then the priority is determined by the interface ID (slot/port). The lowest interface ID is the highest priority. For example, GigabitEthernet 0/0 is a higher priority than GigabitEthernet 0/1.

If you want to prioritize an interface to be active even though it has a higher interface ID, then set this command to have a lower value. For example, to make GigabitEthernet 1/3 active before GigabitEthernet 0/7, then make the **lacp port-priority** value be 12345 on the 1/3 interface vs. the default 32768 on the 0/7 interface.

If the device at the other end of the EtherChannel has conflicting port priorities, the system priority is used to determine which port priorities to use. See the **lacp system-priority** command.

Step 4 (Optional) Set the Ethernet properties for the port-channel interface to override the properties set on the individual interfaces.

interface port-channel *channel_id*

See Enable the Physical Interface and Configure Ethernet Parameters, on page 406 for Ethernet commands. This method provides a shortcut to set these parameters because these parameters must match for all interfaces in the channel group.

Step 5 Repeat Steps 1 through 3 for each interface you want to add to the channel group.

Each interface in the channel group must be the same type and speed. Half duplex is not supported. If you add an interface that does not match, it will be placed in a suspended state.

Related Topics

Link Aggregation Control Protocol, on page 415
Customize the EtherChannel, on page 423

Customize the EtherChannel

This section describes how to set the maximum number of interfaces in the EtherChannel, the minimum number of operating interfaces for the EtherChannel to be active, the load balancing algorithm, and other optional parameters.

Procedure

Step 1 Specify the port-channel interface:

interface port-channel *channel_id*

Example:

```
ciscoasa(config)# interface port-channel 1
```

This interface was created automatically when you added an interface to the channel group. If you have not yet added an interface, then this command creates the port-channel interface.

You need to add at least one member interface to the port-channel interface before you can configure logical parameters for it such as a name.

Step 2 Specify the maximum number of active interfaces allowed in the channel group:

lacp max-bundle *number*

Example:

```
ciscoasa(config-if)# lacp max-bundle 6
```

The *number* is between 1 and 16. The default is 16. If your switch does not support 16 active interfaces, be sure to set this command to 8 or fewer.

Step 3 Specify the minimum number of active interfaces required for the port-channel interface to become active:

port-channel min-bundle *number*

Example:

```
ciscoasa(config-if)# port-channel min-bundle 2
```

The *number* is between 1 and 16. The default is 1. If the active interfaces in the channel group falls below this value, then the port-channel interface goes down, and could trigger a device-level failover.

Step 4 Configure the load-balancing algorithm:

port-channel load-balance {**dst-ip** | **dst-ip-port** | **dst-mac** | **dst-port** | **src-dst-ip** | **src-dst-ip-port** | **src-dst-mac** | **src-dst-port** | **src-ip** | **src-ip-port** | **src-mac** | **src-port** | **vlan-dst-ip** | **vlan-dst-ip-port** | **vlan-only** | **vlan-src-dst-ip** | **vlan-src-dst-ip-port** | **vlan-src-ip** | **vlan-src-ip-port**}

Example:

```
ciscoasa(config-if)# port-channel load-balance src-dst-mac
```

By default, the ASA balances the packet load on interfaces according to the source and destination IP address (**src-dst-ip**) of the packet. If you want to change the properties on which the packet is categorized, use this command. For example, if your traffic is biased heavily towards the same source and destination IP addresses, then the traffic assignment to interfaces in the EtherChannel will be unbalanced. Changing to a different algorithm can result in more evenly distributed traffic.

Step 5 Set the LACP system priority:

lacp system-priority *number*

Example:

```
ciscoasa(config)# lacp system-priority 12345
```

The *number* is between 1 and 65535. The default is 32768. The higher the number, the lower the priority. This command is global for the ASA.

If the device at the other end of the EtherChannel has conflicting port priorities, the system priority is used to determine which port priorities to use. For interface priorities within an EtherChannel, see the **lacp port-priority** command.

Related Topics

Load Balancing, on page 416
Add Interfaces to the EtherChannel, on page 421

Monitoring EtherChannel and Redundant Interfaces

See the following commands:

Note For the Firepower 9300, some statistics are not shown using the ASA commands. You must view more detailed interface statistics using FXOS commands.

- /eth-uplink/fabric# **show interface**
- /eth-uplink/fabric# **show port-channel**
- /eth-uplink/fabric/interface# **show stats**

See the FXOS troubleshooting guide for more information.

- **show interface**

  Displays interface statistics.

- **show interface ip brief**

  Displays interface IP addresses and status.

- **show lacp** {[*channel_group_number*] {**counters** | **internal** | **neighbor**} | **sys-id**}

  For EtherChannel, displays LACP information such as traffic statistics, system identifier and neighbor details.

- **show port-channel** [*channel_group_number*] [**brief** | **detail** | **port** | **protocol** | **summary**]

  For EtherChannel, displays EtherChannel information in a detailed and one-line summary form. This command also displays the port and port-channel information.

- **show port-channel** *channel_group_number* **load-balance** [**hash-result** {**ip** | **ipv6** | **l4port** | **mac** | **mixed** | **vlan-only**} *parameters*]

  For EtherChannel, displays port-channel load-balance information along with the hash result and member interface selected for a given set of parameters.

Examples for EtherChannel and Redundant Interfaces

The following example configures three interfaces as part of an EtherChannel. It also sets the system priority to be a higher priority, and GigabitEthernet 0/2 to be a higher priority than the other interfaces in case more than eight interfaces are assigned to the EtherChannel.

```
lacp system-priority 1234
interface GigabitEthernet0/0
  channel-group 1 mode active
interface GigabitEthernet0/1
  channel-group 1 mode active
interface GigabitEthernet0/2
  lacp port-priority 1234
  channel-group 1 mode passive
interface Port-channel1
  lacp max-bundle 4
  port-channel min-bundle 2
  port-channel load-balance dst-ip
```

History for EtherChannels and Redundant Interfaces

Table 18: History for EtherChannels and Redundant Interfaces

Feature Name	Releases	Feature Information
Redundant interfaces	8.0(2)	A logical redundant interface pairs an active and a standby physical interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the ASA reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as failover if desired. You can configure up to eight redundant interface pairs.
EtherChannel support	8.4(1)	You can configure up to 48 802.3ad EtherChannels of eight active interfaces each. We introduced the following commands: **channel-group**, **lacp port-priority**, **interface port-channel**, **lacp max-bundle**, **port-channel min-bundle**, **port-channel load-balance**, **lacp system-priority**, **clear lacp counters**, **show lacp**, **show port-channel**. **Note** EtherChannel is not supported on the ASA 5505.

Feature Name	Releases	Feature Information
Support for 16 active links in an EtherChannel	9.2(1)	You can now configure up to 16 active links in an EtherChannel. Previously, you could have 8 active links and 8 standby links. Be sure that your switch can support 16 active links (for example the Cisco Nexus 7000 with F2-Series 10 Gigabit Ethernet Module). **Note** If you upgrade from an earlier ASA version, the maximum active interfaces is set to 8 for compatibility purposes (the **lacp max-bundle** command). We modified the following commands: **lacp max-bundle** and **port-channel min-bundle**.

CHAPTER 13

VLAN Subinterfaces

This chapter tells how to configure VLAN subinterfaces.

Note For multiple context mode, complete all tasks in this section in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

About VLAN Subinterfaces

VLAN subinterfaces let you divide a physical, redundant, or EtherChannel interface into multiple logical interfaces that are tagged with different VLAN IDs. An interface with one or more VLAN subinterfaces is automatically configured as an 802.1Q trunk. Because VLANs allow you to keep traffic separate on a given physical interface, you can increase the number of interfaces available to your network without adding additional physical interfaces or ASAs. This feature is particularly useful in multiple context mode so that you can assign unique interfaces to each context.

Licensing for VLAN Subinterfaces

Model	License Requirement
Firepower 9300	Standard License: 1024
ASAv5	Standard License: 25
ASAv10	Standard License: 50

Model	License Requirement
ASAv30	Standard License: 200
ASA 5506-X ASA 5506W-X ASA 5506H-X	Base License: 5 Security Plus License: 30
ASA 5508-X	Base License: 50
ASA 5512-X	Base License: 50 Security Plus License: 100
ASA 5515-X	Base License: 100
ASA 5516-X	Base License: 50
ASA 5525-X	Base License: 200
ASA 5545-X	Base License: 300
ASA 5555-X	Base License: 500
ASA 5585-X	Base and Security Plus License: 1024
ASASM	No support.
ISA 3000	Base License: 5 Security Plus License: 25

Note For an interface to count against the VLAN limit, you must assign a VLAN to it. For example:

```
interface gigabitethernet 0/0.100
  vlan 100
```

Guidelines and Limitations for VLAN Subinterfaces

Model Support

- ASASM—VLAN subinterfaces are not supported on the ASASM; ASASM interfaces are already VLAN interfaces assigned from the switch.
- For most ASA models, you cannot configure subinterfaces on the Management interface. See Management Slot/Port Interface, on page 402 for subinterface support.

Additional Guidelines

- Preventing untagged packets on the physical interface—If you use subinterfaces, you typically do not also want the physical interface to pass traffic, because the physical interface passes untagged packets. This property is also true for the active physical interface in a redundant interface pair and for EtherChannel links. Because the physical, redundant, or EtherChannel interface must be enabled for the subinterface to pass traffic, ensure that the physical, redundant, or EtherChannel interface does not pass traffic by leaving out the **nameif** command. If you want to let the physical, redundant, or EtherChannel interface pass untagged packets, you can configure the **nameif** command as usual.
- The ASA does not support the Dynamic Trunking Protocol (DTP), so you must configure the connected switch port to trunk unconditionally.
- You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

Default Settings for VLAN Subinterfaces

This section lists default settings for interfaces if you do not have a factory default configuration.

Default State of Interfaces

The default state of an interface depends on the type and the context mode.

In multiple context mode, all allocated interfaces are enabled by default, no matter what the state of the interface is in the system execution space. However, for traffic to pass through the interface, the interface also has to be enabled in the system execution space. If you shut down an interface in the system execution space, then that interface is down in all contexts that share it.

In single mode or in the system execution space, interfaces have the following default states:

- Physical interfaces—Disabled.
- VLAN subinterfaces—Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.

Configure VLAN Subinterfaces and 802.1Q Trunking

Add a VLAN subinterface to a physical, redundant, or EtherChannel interface.

Before you begin

For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Step 1 Specify the new subinterface:

interface {*physical_interface* | **redundant** *number* | **port-channel** *number*}.*subinterface*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/1.100
```

The **redundant** *number* argument is the redundant interface ID, such as **redundant 1**.

The **port-channel** *number* argument is the EtherChannel interface ID, such as **port-channel 1**.

The *subinterface* ID is an integer between 1 and 4294967293.

Step 2 Specify the VLAN for the subinterface:

vlan *vlan_id*

Example:

```
ciscoasa(config-subif)# vlan 101
```

The *vlan_id* is an integer between 1 and 4094. Some VLAN IDs might be reserved on connected switches, so check the switch documentation for more information.

You can only assign a single VLAN to a subinterface. You cannot assign the same VLAN to multiple subinterfaces. You cannot assign a VLAN to the physical interface. Each subinterface must have a VLAN ID before it can pass traffic. To change a VLAN ID, you do not need to remove the old VLAN ID with the **no** option; you can enter the **vlan** command with a different VLAN ID, and the ASA changes the old ID.

Related Topics

Licensing for VLAN Subinterfaces, on page 429

Monitoring VLAN Subinterfaces

See the following commands:

- **show interface**

  Displays interface statistics.

- **show interface ip brief**

  Displays interface IP addresses and status.

Examples for VLAN Subinterfaces

The following example configures parameters for a subinterface in single mode:

```
interface gigabitethernet 0/1
  no nameif
  no security-level
  no ip address
  no shutdown
interface gigabitethernet 0/1.1
  vlan 101
  nameif inside
  security-level 100
  ip address 192.168.6.6 255.255.255.0
  no shutdown
```

History for VLAN Subinterfaces

Table 19: History for VLAN Subinterfaces

Feature Name	Version	Feature Information
Increased VLANs	7.0(5)	Increased the following limits: • ASA5510 Base license VLANs from 0 to 10. • ASA5510 Security Plus license VLANs from 10 to 25. • ASA5520 VLANs from 25 to 100. • ASA5540 VLANs from 100 to 200.
Increased VLANs	7.2(2)	VLAN limits were increased for the ASA 5510 (from 10 to 50 for the Base license, and from 25 to 100 for the Security Plus license), the ASA 5520 (from 100 to 150), the ASA 5550 (from 200 to 250).
Increased VLANs for the ASA 5580	8.1(2)	The number of VLANs supported on the ASA 5580 are increased from 100 to 250.

CHAPTER 14

VXLAN Interfaces

This chapter tells how to configure Virtual eXtensible LAN (VXLAN) interfaces. VXLANs act as Layer 2 virtual networks over Layer 3 physical networks to stretch Layer 2 networks.

About VXLAN Interfaces

VXLAN provides the same Ethernet Layer 2 network services as VLAN does, but with greater extensibility and flexibility. Compared to VLAN, VXLAN offers the following benefits:

- Flexible placement of multitenant segments throughout the data center.
- Higher scalability to address more Layer 2 segments: up to 16 million VXLAN segments.

This section describes how VXLAN works. For detailed information, see RFC 7348.

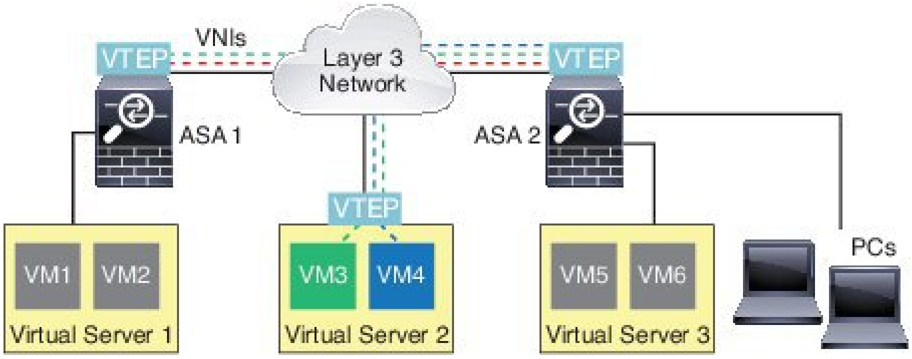
VXLAN Encapsulation

VXLAN is a Layer 2 overlay scheme on a Layer 3 network. It uses MAC Address-in-User Datagram Protocol (MAC-in-UDP) encapsulation. The original Layer 2 frame has a VXLAN header added and is then placed in a UDP-IP packet.

VXLAN Tunnel Endpoint

VXLAN tunnel endpoint (VTEP) devices perform VXLAN encapsulation and decapsulation. Each VTEP has two interface types: one or more virtual interfaces called VXLAN Network Identifier (VNI) interfaces to which you apply your security policy, and a regular interface called the VTEP source interface that tunnels the VNI interfaces between VTEPs. The VTEP source interface is attached to the transport IP network for VTEP-to-VTEP communication.

The following figure shows two ASAs and Virtual Server 2 acting as VTEPs across a Layer 3 network, extending the VNI 1, 2, and 3 networks between sites. The ASAs act as bridges or gateways between VXLAN and non-VXLAN networks.



The underlying IP network between VTEPs is independent of the VXLAN overlay. Encapsulated packets are routed based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address. The destination IP address can be a multicast group when the remote VTEP is not known. The destination port is UDP port 4789 by default (user configurable).

VTEP Source Interface

The VTEP source interface is a regular ASA interface (physical, redundant, EtherChannel, or even VLAN) with which you plan to associate all VNI interfaces. You can configure one VTEP source interface per ASA/security context.

The VTEP source interface can be devoted wholly to VXLAN traffic, although it is not restricted to that use. If desired, you can use the interface for regular traffic and apply a security policy to the interface for that traffic. For VXLAN traffic, however, all security policy must be applied to the VNI interfaces. The VTEP interface serves as a physical port only.

In transparent firewall mode, the VTEP source interface is not part of a BVI, and you do configure an IP address for it, similar to the way the management interface is treated.

VNI Interfaces

VNI interfaces are similar to VLAN interfaces: they are virtual interfaces that keep network traffic separated on a given physical interface by using tagging. You apply your security policy directly to each VNI interface.

All VNI interfaces are associated with the same VTEP interface.

VXLAN Packet Processing

Traffic entering and exiting the VTEP source interface is subject to VXLAN processing, specifically encapsulation or decapsulation.

Encapsulation processing includes the following tasks:

- The VTEP source interface encapsulates the inner MAC frame with the VXLAN header.
- The UDP checksum field is set to zero.

- The Outer frame source IP is set to the VTEP interface IP.
- The Outer frame destination IP is decided by a remote VTEP IP lookup.

Decapsulation; the ASA only decapsulates a VXLAN packet if:

- It is a UDP packet with the destination port set to 4789 (this value is user configurable).
- The ingress interface is the VTEP source interface.
- The ingress interface IP address is the same as the destination IP address.
- The VXLAN packet format is compliant with the standard.

Peer VTEPs

When the ASA sends a packet to a device behind a peer VTEP, the ASA needs two important pieces of information:

- The destination MAC address of the remote device
- The destination IP address of the peer VTEP

There are two ways in which the ASA can find this information:

- A single peer VTEP IP address can be statically configured on the ASA.

  You cannot manually define multiple peers.

  The ASA then sends a VXLAN-encapsulated ARP broadcast to the VTEP to learn the end node MAC address.
- A multicast group can be configured on each VNI interface (or on the VTEP as a whole).

  The ASA sends a VXLAN-encapsulated ARP broadcast packet within an IP multicast packet through the VTEP source interface. The response to this ARP request enables the ASA to learn both the remote VTEP IP address along with the destination MAC address of the remote end node.

The ASA maintains a mapping of destination MAC addresses to remote VTEP IP addresses for the VNI interfaces.

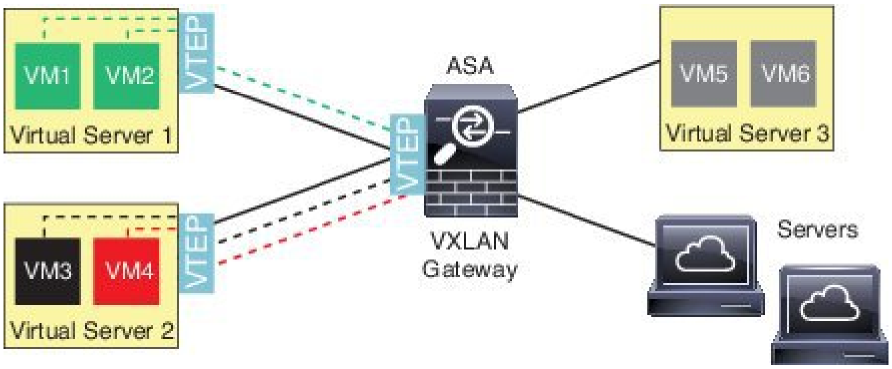
VXLAN Use Cases

This section describes the use cases for implementing VXLAN on the ASA.

VXLAN Bridge or Gateway Overview

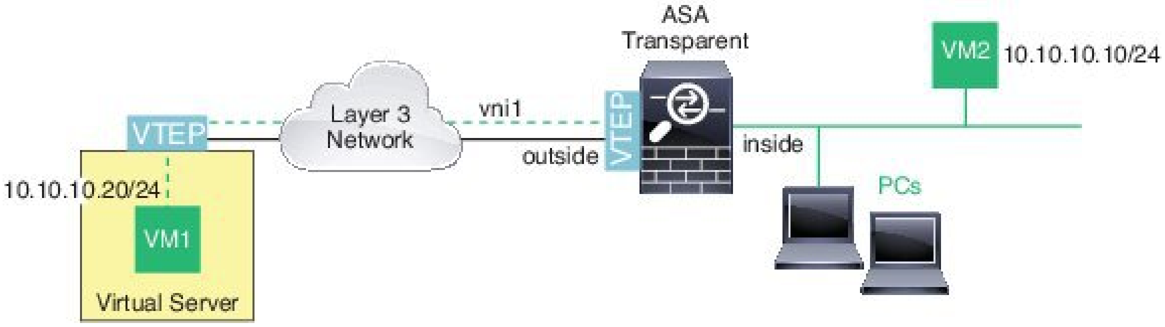
Each ASA VTEP acts as a bridge or gateway between end nodes such as VMs, servers, and PCs and the VXLAN overlay network. For incoming frames received with VXLAN encapsulation over the VTEP source interface, the ASA strips out the VXLAN header and forwards it to a physical interface connected to a non-VXLAN network based on the destination MAC address of the inner Ethernet frame.

The ASA always processes VXLAN packets; it does not just forward VXLAN packets untouched between two other VTEPs.



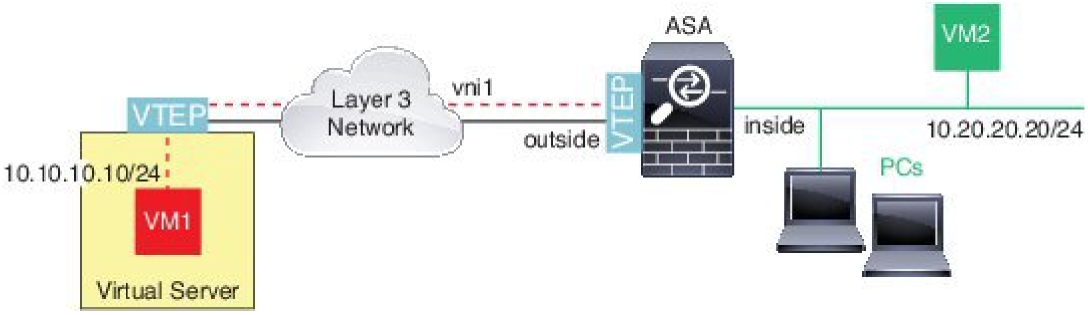
VXLAN Bridge (Transparent Mode)

When you use a bridge group (transparent firewall mode), the ASA can serve as a VXLAN bridge between a (remote) VXLAN segment and a local segment where both are in the same network. In this case, one member of the bridge group is a regular interface while the other member is a VNI interface.



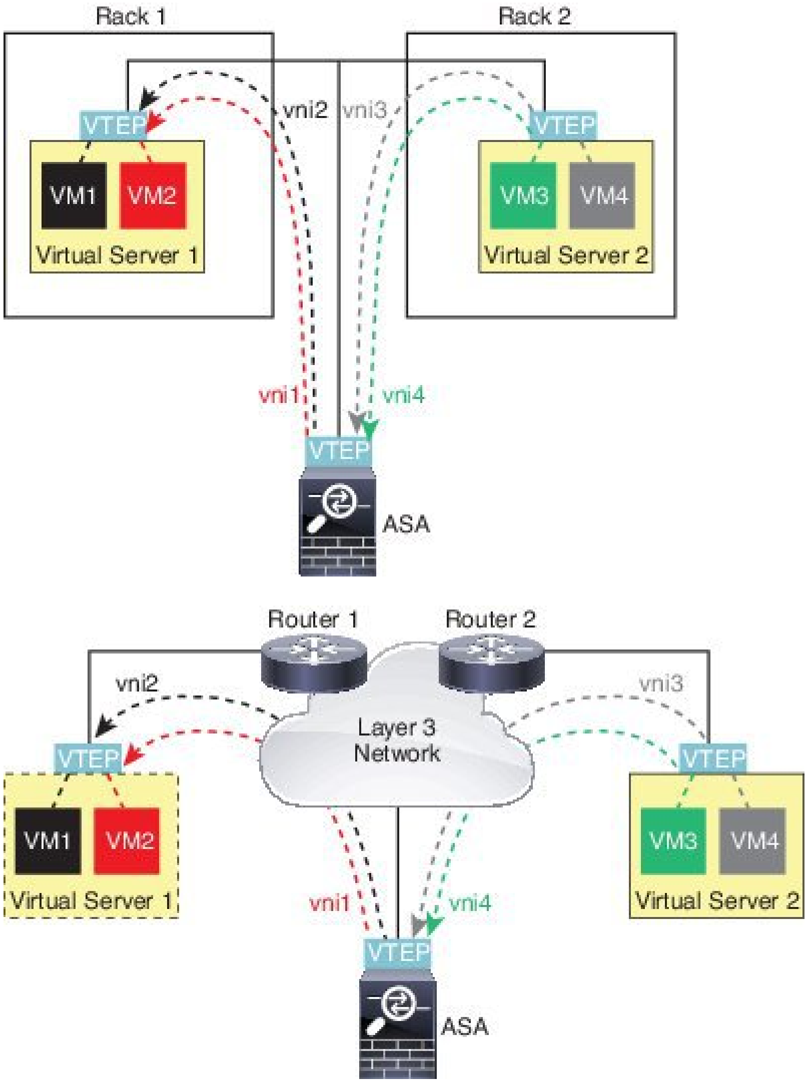
VXLAN Gateway (Routed Mode)

The ASA can serve as a router between VXLAN and non-VXLAN domains, connecting devices on different networks.



Router Between VXLAN Domains

With a VXLAN-stretched Layer 2 domain, a VM can point to an ASA as its gateway while the ASA is not on the same rack, or even when the ASA is far away over the Layer 3 network.



See the following notes about this scenario:

1. For packets from VM3 to VM1, the destination MAC address is the ASA MAC address, because the ASA is the default gateway.
2. The VTEP source interface on Virtual Server 2 receives packets from VM3, then encapsulates the packets with VNI 3's VXLAN tag and sends them to the ASA.
3. When the ASA receives the packets, it decapsulates the packets to get the inner frames.
4. The ASA uses the inner frames for route lookup, then finds that the destination is on VNI 2. If it does not already have a mapping for VM1, the ASA sends an encapsulated ARP broadcast on the multicast group IP on VNI 2.

Note The ASA must use dynamic VTEP peer discovery because it has multiple VTEP peers in this scenario.

5. The ASA encapsulates the packets again with the VXLAN tag for VNI 2 and sends the packets to Virtual Server 1. Before encapsulation, the ASA changes the inner frame destination MAC address to be the MAC of VM1 (multicast-encapsulated ARP might be needed for the ASA to learn the VM1 MAC address).
6. When Virtual Server 1 receives the VXLAN packets, it decapsulates the packets and delivers the inner frames to VM1.

Guidelines for VXLAN Interfaces

IPv6

- The VNI interface supports IPv6 traffic, but the VTEP source interface IP address only supports IPv4.
- IPv6 OSPF interface settings are not supported.

Clustering

ASA clustering does not support VXLAN in Individual Interface mode. Only Spanned EtherChannel mode supports VXLAN.

Routing

- Only static routing is supported on the VNI interface; dynamic routing protocols are not supported.
- Policy Based Routing is not supported.

MTU

If the source interface MTU is less than 1554 bytes, then the ASA automatically raises the MTU to 1554 bytes. In this case, the entire Ethernet datagram is being encapsulated, so the new packet is larger and requires a larger MTU. If the MTU used by other devices is larger, then you should set the source interface MTU to be the network MTU + 54 bytes. This MTU requires you to enable jumbo frame reservation; see Enable Jumbo Frame Support (ASA Models), on page 408.

Default Settings for VXLAN Interfaces

VNI interfaces are enabled by default.

Configure VXLAN Interfaces

To configure VXLAN, perform the following steps.

Procedure

Step 1 Configure the VTEP Source Interface, on page 441.

Step 2 Configure the VNI Interface, on page 442

Step 3 (Optional) Change the VXLAN UDP Port, on page 444.

Configure the VTEP Source Interface

You can configure one VTEP source interface per ASA or per security context. The VTEP is defined as a Network Virtualization Endpoint (NVE); VXLAN VTEP is the only supported NVE at this time.

Before you begin

For multiple context mode, complete the tasks in this section in the context execution space. Enter the **changeto context** *name* command to change to the context you want to configure.

Procedure

Step 1 (Transparent mode) Specify that the source interface is NVE-only:

interface *id*

nve-only

Example:

```
ciscoasa(config)# interface gigabitethernet 1/1
ciscoasa(config-if)# nve-only
```

This setting lets you configure an IP address for the interface. This command is optional for routed mode where this setting restricts traffic to VXLAN and common management traffic only on this interface.

Step 2 Configure the source interface name and IPv4 address.

Example:

(Routed Mode)

```
ciscoasa(config)# interface gigabitethernet 1/1
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
```

Example:

(Transparent Mode)

```
ciscoasa(config)# interface gigabitethernet 1/1
ciscoasa(config-if)# nve-only
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
```

Step 3 Specify the NVE instance:

nve 1

You can only specify one NVE instance, with the ID 1.

Note The **encapsulation vxlan** command is added by default for the NVE instance; you do not need to explicitly add it.

Step 4 Specify the source interface name that you configured in Step 2:

source-interface *interface-name*

Example:

```
ciscoasa(cfg-nve)# source-interface outside
```

Note If the source interface MTU is less than 1554 bytes, then the ASA automatically raises the MTU to 1554 bytes.

Step 5 (Multiple context mode; Optional for single mode) Manually specify the peer VTEP IP address:

peer ip *ip_address*

Example:

```
ciscoasa(cfg-nve)# peer ip 10.1.1.2
```

If you specify the peer IP address, you cannot use multicast group discovery. Multicast is not supported in multiple context mode, so manual configuration is the only option. You can only specify one peer for the VTEP.

Step 6 (Optional; single mode only) Specify a default multicast group for all associated VNI interfaces:

default-mcast-group *mcast_ip*

Example:

```
ciscoasa(cfg-nve)# default-mcast-group 236.0.0.100
```

If you do not configure the multicast group per VNI interface, then this group is used. If you configure a group at the VNI interface level, then that group overrides this setting.

Configure the VNI Interface

Add a VNI interface, associate it with the VTEP source interface, and configure basic interface parameters.

Procedure

Step 1 Create the VNI interface:

interface vni *vni_num*

Example:

```
ciscoasa(config)# interface vni 1
```

Set the ID between 1 and 10000. This ID is only an internal interface identifier.

Step 2 Specify the VXLAN segment ID:

segment-id *id*

Example:

```
ciscoasa(config-if)# segment-id 1000
```

Set the ID between 1 and 16777215. The segment ID is used for VXLAN tagging.

Step 3 (Required for transparent mode) Specify the bridge group to which you want to associate this interface:

bridge-group *number*

Example:

```
ciscoasa(config-if)# bridge-group 1
```

See Configure Transparent Mode Bridge Group Interfaces, on page 458 to configure the BVI interface and associate regular interfaces to this bridge group.

Step 4 Associate this interface with the VTEP source interface:

vtep-nve 1

Step 5 Name the interface:

nameif *vni_interface_name*

Example:

```
ciscoasa(config-if)# nameif vxlan1000
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 6 (Routed mode) Assign an IPv4 and/or IPv6 address:

ip address {*ip_address* [*mask*] [**standby** *ip_address*] | **dhcp** [**setroute**] | **pppoe** [**setroute**]}

ipv6 address {**autoconfig** | *ipv6-address/prefix-length* [**standby** *ipv6-address*]}

Example:

```
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
ciscoasa(config-if)# ipv6 address 2001:0DB8::BA98:0:3210/48
```

Step 7 Set the security level:

security-level *level*

Example:

```
ciscoasa(config-if)# security-level 50
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 8 (Single mode) Set the multicast group address:

mcast-group *multicast_ip*

Example:

```
ciscoasa(config-if)# mcast-group 236.0.0.100
```

If you do not set the multicast group for the VNI interface, the default group from the VTEP source interface configuration is used, if available. If you manually set a VTEP peer IP for the VTEP source interface, you cannot specify a multicast group for the VNI interface. Multicast is not supported in multiple context mode.

(Optional) Change the VXLAN UDP Port

By default, the VTEP source interface accepts VXLAN traffic to UDP port 4789. If your network uses a non-standard port, you can change it.

Before you begin

For multiple context mode, complete this task in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Set the VXLAN UDP port:

vxlan *port number*

Example:

```
ciscoasa(config)# vxlan port 5678
```

Monitoring VXLAN Interfaces

See the following commands to monitor VTEP and VNI interfaces.

- **show nve** [*id*] [**summary**]

  This command shows the parameters, status and statistics of a NVE interface, status of its carrier interface (source-interface), IP address of the carrier interface, VNIs that use this NVE as the VXLAN VTEP, and peer VTEP IP addresses associated with this NVE interface. With the **summary** option, this command only shows the status of the NVE interface, number of VNIs behind the NVE interface, and number of VTEPs discovered.

  See the following output for the **show nve 1** command:

```
ciscoasa# show nve 1
ciscoasa(config-if)# show nve
nve 1, source-interface "inside" is up
IP address 15.1.2.1, subnet mask 255.255.255.0
Encapsulation: vxlan
Encapsulated traffic statistics:
6701004 packets input, 3196266002 bytes
6700897 packets output, 3437418084 bytes
1 packets dropped
Number of configured static peer VTEPs: 0
Number of discovered peer VTEPs: 1
Discovered peer VTEPs:
IP address 15.1.2.3
Number of VNIs attached to nve 1: 2
VNIs attached:
vni 2: segment-id 5002, mcast-group 239.1.2.3
vni 1: segment-id 5001, mcast-group 239.1.2.3
```

See the following output for the **show nve 1 summary** command:

```
ciscoasa# show nve 1 summary
nve 1, source-interface "inside" is up
Encapsulation: vxlan
Number of configured static peer VTEPs: 0
Number of discovered peer VTEPs: 1
Default multicast group: 239.1.2.3
Number of VNIs attached to nve 1: 2
```

- **show interface vni** *id* [**summary**]

  This command shows the parameters, status and statistics of a VNI interface, status of its bridged interface (if configured), and NVE interface it is associated with. The **summary** option shows only the VNI interface parameters.

  See the following output for the **show interface vni 1** command:

```
ciscoasa# show interface vni 1
Interface vni1 "vni-inside", is up, line protocol is up
VTEP-NVE 1
Segment-id 5001
Tag-switching: disabled
MTU: 1500
MAC: aaaa.bbbb.1234
IP address 192.168.0.1, subnet mask 255.255.255.0
Multicast group 239.1.3.3
Traffic Statistics for "vni-inside":
235 packets input, 23606 bytes
524 packets output, 32364 bytes
14 packets dropped
1 minute input rate 0 pkts/sec, 0 bytes/sec
1 minute output rate 0 pkts/sec, 2 bytes/sec
1 minute drop rate, 0 pkts/sec
5 minute input rate 0 pkts/sec, 0 bytes/sec
5 minute output rate 0 pkts/sec, 0 bytes/sec
5 minute drop rate, 0 pkts/sec
```

  See the following output for the **show interface vni 1 summary** command:

```
ciscoasa# show interface vni 1 summary
Interface vni1 "vni-inside", is up, line protocol is up
VTEP-NVE 1
Segment-id 5001
Tag-switching: disabled
MTU: 1500
MAC: aaaa.bbbb.1234
IP address 192.168.0.1, subnet mask 255.255.255.0
Multicast group not configured
```

- **show vni vlan-mapping**

  This command shows the mapping between VNI segment IDs and VLAN interfaces or physical interfaces. This command is only valid in transparent firewall mode because in routed mode, the mapping between VXLANs and VLANs can include too many values to show.

  See the following output for the **show vni vlan-mapping** command:

```
ciscoasa# show vni vlan-mapping
vni1: segment-id: 6000, interface: 'g0110', vlan 10, interface: 'g0111', vlan 11
vni2: segment_id: 5000, interface: 'g01100', vlan 1, interface: 'g111', vlan 3, interface:
 'g112', vlan 4
```

- **show arp vtep-mapping**

  This command displays MAC addresses cached on the VNI interface for IP addresses located in the remote segment domain and the remote VTEP IP addresses.

  See the following output for the **show arp vtep-mapping** command:

```
ciscoasa# show arp vtep-mapping
vni-outside 192.168.1.4 0012.0100.0003 577 15.1.2.3
vni-inside 192.168.0.4 0014.0100.0003 577 15.1.2.3
```

- **show mac-address-table vtep-mapping**

  This command displays the Layer 2 forwarding table (MAC address table) on the VNI interface with the remote VTEP IP addresses.

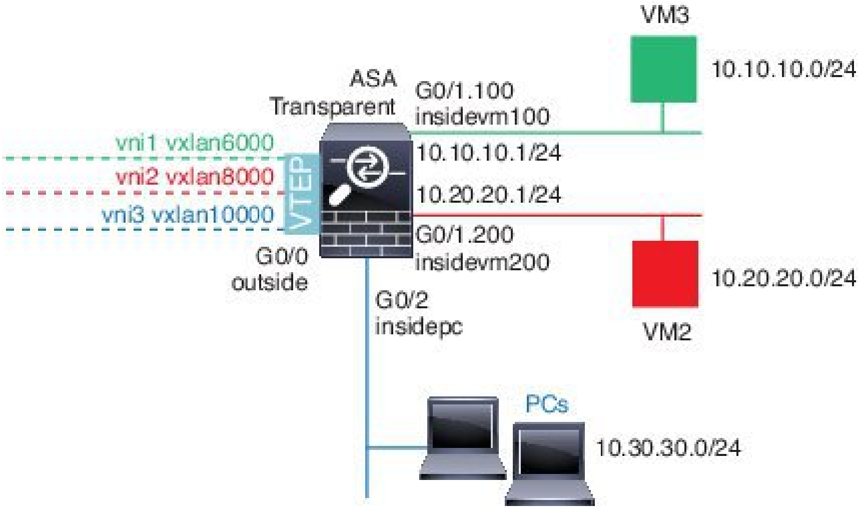
  See the following output for the **show mac-address-table vtep-mapping** command:

```
ciscoasa# show mac-address-table vtep-mapping
interface                  mac  address       type     Age(min)   bridge-group    VTEP
--------------------------------------------------------------------------------------
vni-outside               00ff.9200.0000      dynamic      5         1        10.9.1.3
vni-inside                 0041.9f00.0000      dynamic      5         1      10.9.1.3
```

Examples for VXLAN Interfaces

See the following configuration examples for VXLAN.

Transparent VXLAN Gateway Example



See the following description of this example:

- The outside interface on GigabitEthernet 0/0 is used as the VTEP source interface, and it is connected to the Layer 3 network.
- The insidevm100 VLAN subinterface on GigabitEthernet 0/1.100 is connected to the 10.10.10.0/24 network, on which VM3 resides. When VM3 communicates with VM1 (not shown; both have 10.10.10.0/24 IP addresses), the ASA uses VXLAN tag 6000.
- The insidevm200 VLAN subinterface on GigabitEthernet 0/1.200 is connected to the 10.20.20.0/24 network, on which VM2 resides. When VM2 communicates with VM4 (not shown; both have 10.20.20.0/24 IP addresses), the ASA uses VXLAN tag 8000.
- The insidepc interface on GigabitEthernet 0/2 is connected to the 10.30.30.0/24 network on which a few PCs reside. When those PCs communicate with VMs/PCs (not shown) behind a remote VTEP that belongs to same network (all have 10.30.30.0/24 IP addresses), the ASA uses VXLAN tag 10000.

ASA Configuration

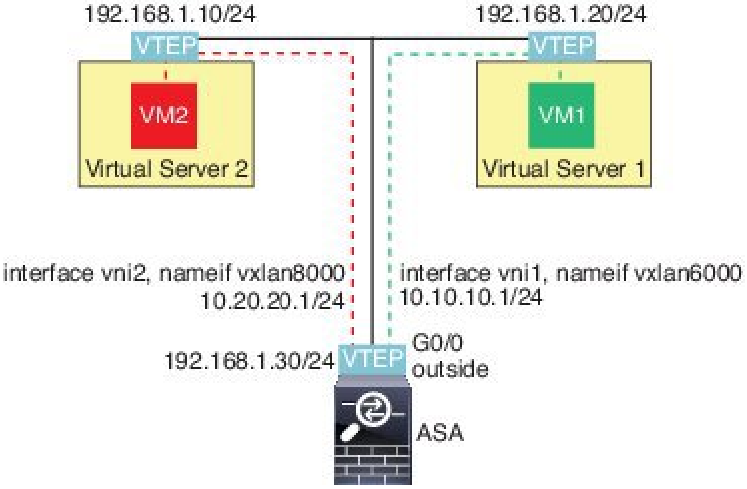
```
firewall transparent
vxlan port 8427
!
interface gigabitethernet0/0
  nve-only
  nameif outside
  ip address 192.168.1.30 255.255.255.0
  no shutdown
!
nve 1
  encapsulation vxlan
  source-interface outside
!
interface vni1
  segment-id 6000
  nameif vxlan6000
  security-level 0
  bridge-group 1
  vtep-nve 1
  mcast-group 235.0.0.100
!
interface vni2
  segment-id 8000
  nameif vxlan8000
  security-level 0
  bridge-group 2
  vtep-nve 1
  mcast-group 236.0.0.100
!
interface vni3
  segment-id 10000
  nameif vxlan10000
  security-level 0
  bridge-group 3
  vtep-nve 1
  mcast-group 236.0.0.100
!
interface gigabitethernet0/1.100
  nameif insidevm100
  security-level 100
  bridge-group 1
!
interface gigabitethernet0/1.200
  nameif insidevm200
  security-level 100
  bridge-group 2
!
interface gigabitethernet0/2
  nameif insidepc
  security-level 100
  bridge-group 3
!
interface bvi 1
  ip address 10.10.10.1 255.255.255.0
!
interface bvi 2
  ip address 10.20.20.1 255.255.255.0
!
interface bvi 3
  ip address 10.30.30.1 255.255.255.0
```

Notes

- For VNI interfaces vni1 and vni2, the inner VLAN tag is removed during encapsulation.
- VNI interfaces vni2 and vni3 share the same multicast IP address for encapsulated ARP over multicast. This sharing is allowed.
- The ASA bridges the VXLAN traffic to non-VXLAN-supported interfaces based on the above BVIs and bridge group configurations. For each of the stretched Layer 2 network segments (10.10.10.0/24, 10.20.20.0/24 and 10.30.30.0/24), the ASA serves as a bridge.
- It is allowed to have more than one VNI or more than one regular interface (VLAN or just physical interface) in a bridge group. The forwarding or association between VXLAN segment ID to the VLAN ID (or a physical interface) is decided by the destination MAC address and which interface connects to the destination.

- The VTEP source-interface is a Layer 3 interface in transparent firewall mode indicated by **nve-only** in the interface configuration. The VTEP source interface is not a BVI interface or a management interface, but it has an IP address and uses the routing table.

VXLAN Routing Example



See the following description of this example:

- VM1 (10.10.10.10) is hosted on Virtual Server 1, and VM2 (10.20.20.20) is hosted on Virtual Server 2.
- The default gateway for VM1 is the ASA, which is not in the same pod as Virtual Server 1, but VM1 is not aware of it. VM1 only knows that its default gateway IP address is 10.10.10.1. Similarly, VM2 only knows that its default gateway IP address is 10.20.20.1.
- The VTEP-supported hypervisors on Virtual Server 1 and 2 are able to communicate with the ASA over the same subnet or through a Layer 3 network (not shown; in which case, the ASA and uplinks of virtual servers have different network addresses).
- VM1's packet will be encapsulated by its hypervisor's VTEP and sent to its default gateway over VXLAN tunneling.
- When VM1 sends a packet to VM2, the packet will be sent through default gateway 10.10.10.1 from its perspective. Virtual Server1 knows 10.10.10.1 is not local, so the VTEP encapsulates the packet over VXLAN and sends it to ASA's VTEP.
- On the ASA, the packet is decapsulated. The VXLAN segment ID is learned during decapsulation. The ASA then re-injects the inner frame to the corresponding VNI interface (vni1) based on the VXLAN segment ID. The ASA then conducts a route lookup and sends the inner packet through another VNI interface, vni2. All egressing packets through vni2 are encapsulated with VXLAN segment 8000 and sent through the VTEP to outside.
- Eventually the encapsulated packet is received by the VTEP of Virtual Server 2, which decapsulates it and forwards it to VM2.

ASA Configuration

```
interface gigabitethernet0/0
  nameif outside
  ip address 192.168.1.30 255.255.255.0
  no shutdown
!
nve 1
  encapsulation vxlan
  source-interface outside
  default-mcast-group 235.0.0.100
!
interface vni1
  segment-id 6000
  nameif vxlan6000
  security-level 0
  vtep-nve 1
  ip address 10.20.20.1 255.255.255.0
!
interface vni2
  segment-id 8000
  nameif vxlan8000
  security-level 0
  vtep-nve 1
  ip address 10.10.10.1 255.255.255.0
!
```

History for VXLAN Interfaces

Table 20: History for VXLAN Interfaces

Feature Name	Releases	Feature Information
VXLAN support	9.4(1)	VXLAN support was added, including VXLAN tunnel endpoint (VTEP) support. You can define one VTEP source interface per ASA or security context. We introduced the following commands: **debug vxlan**, **default-mcast-group**, **encapsulation vxlan**, **inspect vxlan**, **interface vni**, **mcast-group**, **nve**, **nve-only**, **peer ip**, **segment-id**, **show arp vtep-mapping**, **show interface vni**, **show mac-address-table vtep-mapping**, **show nve**, **show vni vlan-mapping**, **source-interface**, **vtep-nve**, **vxlan port**

CHAPTER 15

Routed and Transparent Mode Interfaces

This chapter includes tasks to complete the interface configuration for all models in routed or transparent firewall mode.

Note For multiple context mode, complete the tasks in this section in the context execution space. Enter the **changeto context** *name* command to change to the context you want to configure.

About Routed and Transparent Mode Interfaces

The ASA supports two types of interfaces: routed and bridged.

Each Layer 3 routed interface requires an IP address on a unique subnet.

Bridged interfaces belong to a bridge group, and all interfaces are on the same network. The bridge group is represented by a Bridge Virtual Interface (BVI) that has an IP address on the bridge network. Routed mode only supports routed interfaces. Transparent firewall mode only supports bridge group and BVI interfaces.

Security Levels

Each interface must have a security level from 0 (lowest) to 100 (highest), including bridge group member interfaces. For example, you should assign your most secure network, such as the inside host network, to level 100. While the outside network connected to the Internet can be level 0. Other networks, such as DMZs can be in between. You can assign interfaces to the same security level.

In transparent mode, the BVI interface does not have a security level because it does not participate in routing between interfaces.

The level controls the following behavior:

- Network access—By default, there is an implicit permit from a higher security interface to a lower security interface (outbound). Hosts on the higher security interface can access any host on a lower security interface. You can limit access by applying an ACL to the interface.

  If you enable communication for same-security interfaces, there is an implicit permit for interfaces to access other interfaces on the same security level or lower.

- Inspection engines—Some application inspection engines are dependent on the security level. For same-security interfaces, inspection engines apply to traffic in either direction.
  - NetBIOS inspection engine—Applied only for outbound connections.
  - SQL*Net inspection engine—If a control connection for the SQL*Net (formerly OraServ) port exists between a pair of hosts, then only an inbound data connection is permitted through the ASA.

Dual IP Stack (IPv4 and IPv6)

The ASA supports both IPv6 and IPv4 addresses on an interface. Make sure you configure a default route for both IPv4 and IPv6.

Guidelines and Limitations for Routed and Transparent Mode Interfaces

Context Mode

- In multiple context mode, you can only configure context interfaces that you already assigned to the context in the system configuration according to Configure Multiple Contexts, on page 185.
- PPPoE is not supported in multiple context mode.
- For multiple context mode in transparent mode, each context must use different interfaces; you cannot share an interface across contexts.
- For multiple context mode in transparent mode, each context typically uses a different subnet. You can use overlapping subnets, but your network topology requires router and NAT configuration to make it possible from a routing standpoint.

Failover

- Do not configure failover links with the procedures in this chapter. See the Failover chapter for more information.
- When you use Failover, you must set the IP address and standby address for data interfaces manually; DHCP and PPPoE are not supported.

IPv6

- IPv6 is supported on all interfaces.

- You can only configure IPv6 addresses manually in transparent mode.
- The ASA does not support IPv6 anycast addresses.

VLAN IDs for the ASASM

You can add any VLAN ID to the configuration, but only VLANs that are assigned to the ASA by the switch can pass traffic. To view all VLANs assigned to the ASA, use the **show vlan** command.

If you add an interface for a VLAN that is not yet assigned to the ASA by the switch, the interface will be in the down state. When you assign the VLAN to the ASA, the interface changes to an up state. See the **show interface** command for more information about interface states.

Transparent Mode and Bridge Group Guidelines

- You can create up to 250 bridge groups, with 4 interfaces per bridge group.
- Each directly-connected network must be on the same subnet.
- The ASA does not support traffic on secondary networks; only traffic on the same network as the BVI IP address is supported.
- An IP address for the BVI is required for each bridge group for to-the-device and from-the-device management traffic, as well as for data traffic to pass through the ASA. For IPv4 traffic, specify an IPv4 address. For IPv6 traffic, specify an IPv6 address.
- You can only configure IPv6 addresses manually.
- The BVI IP address must be on the same subnet as the connected network. You cannot set the subnet to a host subnet (255.255.255.255).
- Management interfaces are not supported as bridge group members.
- In transparent mode, you must use at least 1 bridge group; data interfaces must belong to a bridge group.
- In transparent mode, do not specify the BVI IP address as the default gateway for connected devices; devices need to specify the router on the other side of the ASA as the default gateway.
- In transparent mode, the *default* route, which is required to provide a return path for management traffic, is only applied to management traffic from one bridge group network. This is because the default route specifies an interface in the bridge group as well as the router IP address on the bridge group network, and you can only define one default route. If you have management traffic from more than one bridge group network, you need to specify a regular static route that identifies the network from which you expect management traffic.
- In transparent mode, PPPoE is not supported for the Management interface.
- Bidirectional Forwarding Detection (BFD) echo packets are not allowed through the ASA when using bridge group members. If there are two neighbors on either side of the ASA running BFD, then the ASA will drop BFD echo packets because they have the same source and destination IP address and appear to be part of a LAND attack.

Default Security Level

The default security level is 0. If you name an interface “inside,” and you do not set the security level explicitly, then the ASA sets the security level to 100.

Note If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the connections using the **clear conn** command.

Additional Guidelines and Requirements

- The ASA supports only one 802.1Q header in a packet and does not support multiple headers (known as Q-in-Q support).

Configure Routed Mode Interfaces

To configure routed mode interfaces, perform the following steps.

Configure General Routed Mode Interface Parameters

This procedure describes how to set the name, security level, IPv4 address, and other options.

Before you begin

In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Enter interface configuration mode:

interface *id*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

The interface ID can be:

- **redundant**
- **port-channel**
- *physical*—For example, **ethernet**, **gigabitethernet**, **tengigabitethernet**, **management**. Refer to the hardware installation guide for your model for interface names.
- *physical***.***subinterface*—For example, **gigabitethernet0/0.100**.
- **vni**
- **vlan**
- *mapped_name*—For multiple context mode.

Step 2 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif inside
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 3 Set the IP address using one of the following methods.

For failover and clustering, you must set the IP address manually; DHCP and PPPoE are not supported.

- Set the IP address manually:

  ip address *ip_address* [*mask*] [**standby** *ip_address*]

  Example:

  ```
  ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
  ```

  The standby *ip_address* argument is used for failover. If you do not set the standby IP address, the active unit cannot monitor the standby interface using network tests; it can only track the link state.

  The *ip_address* and *mask* arguments set the interface IP address and subnet mask.

- Obtain an IP address from a DHCP server:

  ip address dhcp [**setroute**]

  Example:

  ```
  ciscoasa(config-if)# ip address dhcp
  ```

  The **setroute** keyword lets the ASA use the default route supplied by the DHCP server.

  Reenter this command to reset the DHCP lease and request a new lease.

  Note If you do not enable the interface using the **no shutdown** command before you enter the **ip address dhcp** command, some DHCP requests might not be sent.

- Obtain an IP address from a PPPoE server:

  ip address pppoe [**setroute**]

  Example:

  ```
  ciscoasa(config-if)# ip address pppoe setroute
  ```

  You can alternatively enable PPPoE by manually entering the IP address:

  ip address *ip_address mask* **pppoe**

  Example:

```
ciscoasa(config-if)# ip address 10.1.1.78 255.255.255.0 pppoe
```

The **setroute** option sets the default routes when the PPPoE client has not yet established a connection. When using the **setroute** option, you cannot have a statically defined route in the configuration.

Note If PPPoE is enabled on two interfaces (such as a primary and backup interface), and you do not configure dual ISP support, then the ASA can only send traffic through the first interface to acquire an IP address.

Step 4 Set the security level:

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 50
```

The *number* is an integer between 0 (lowest) and 100 (highest)..

Step 5 (Optional) Set an interface to management-only mode so that it does not pass through traffic:

management-only

By default, Management interfaces are configured as management-only.

Examples

The following example configures parameters for VLAN 101:

```
ciscoasa(config)# interface vlan 101
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
```

The following example configures parameters in multiple context mode for the context configuration. The interface ID is a mapped name.

```
ciscoasa/contextA(config)# interface int1
ciscoasa/contextA(config-if)# nameif outside
ciscoasa/contextA(config-if)# security-level 100
ciscoasa/contextA(config-if)# ip address 10.1.2.1 255.255.255.0
```

Related Topics

Configure IPv6 Addressing, on page 462
Enable the Physical Interface and Configure Ethernet Parameters, on page 406
Configure PPPoE, on page 457

Configure PPPoE

If the interface is connected to a DSL, cable modem, or other connection to your ISP, and your ISP uses PPPoE to provide your IP address, configure the following parameters.

Procedure

Step 1 Define the Virtual Private Dialup Network (VPDN) group name of your choice to represent this connection:

vpdn group *group_name* **request dialout pppoe**

Example:

```
ciscoasa(config)# vpdn group pppoe-sbc request dialout pppoe
```

Step 2 If your ISP requires authentication, select an authentication protocol:

vpdn group *group_name* **ppp authentication** {**chap** | **mschap** | **pap**}

Example:

```
ciscoasa(config)# vpdn group pppoe-sbc ppp authentication chap
```

Enter the appropriate keyword for the type of authentication used by your ISP.

When using CHAP or MS-CHAP, the username may be referred to as the remote system name, while the password may be referred to as the CHAP secret.

Step 3 Associate the username assigned by your ISP to the VPDN group:

vpdn group *group_name* **localname** *username*

Example:

```
ciscoasa(config)# vpdn group pppoe-sbc localname johncrichton
```

Step 4 Create a username and password pair for the PPPoE connection:

vpdn username *username* **password** *password* [**store-local**]

Example:

```
ciscoasa(config)# vpdn username johncrichton password moya
```

The **store-local** option stores the username and password in a special location of NVRAM on the ASA. If an Auto Update Server sends a **clear config** command to the ASA and the connection is then interrupted, the ASA can read the username and password from NVRAM and re-authenticate to the Access Concentrator.

Configure Transparent Mode Bridge Group Interfaces

A bridge group is a group of interfaces that the ASA bridges instead of routes. Bridge groups are only supported in Transparent Firewall Mode. For more information about bridge groups, see About Bridge Groups, on page 150.

To configure bridge groups and associated interfaces, perform these steps.

Configure the Bridge Virtual Interface (BVI)

Each bridge group requires a BVI for which you configure an IP address. The ASA uses this IP address as the source address for packets originating from the bridge group. The BVI IP address must be on the same subnet as the connected network. For IPv4 traffic, the BVI IP address is required to pass any traffic. For IPv6 traffic, you must, at a minimum, configure the link-local addresses to pass traffic, but a global management address is recommended for full functionality, including remote management and other management operations.

Some models include a bridge group and BVI in the default configuration. You can create additional bridge groups and BVIs and reassign member interfaces between the groups.

Note For a separate management interface in transparent mode (for supported models), a non-configurable bridge group (ID 301) is automatically added to your configuration. This bridge group is not included in the bridge group limit.

Procedure

Step 1 Create a BVI:

interface bvi *bridge_group_number*

Example:

```
ciscoasa(config)# interface bvi 2
```

The *bridge_group_number* is an integer between 1 and 250. You will later assign physical interfaces to this bridge group number.

Step 2 Specify the IP address for the BVI:

ip address *ip_address* [*mask*] [**standby** *ip_address*]

Example:

```
ciscoasa(config-if)# ip address 10.1.3.1 255.255.255.0 standby 10.1.3.2
```

Do not assign a host address (/32 or 255.255.255.255) to the BVI. Also, do not use other subnets that contain fewer than 3 host addresses (one each for the upstream router, downstream router, and BVI) such as a /30 subnet (255.255.255.252). The ASA drops all ARP packets to or from the first and last addresses in a subnet.

Therefore, if you use a /30 subnet and assign a reserved address from that subnet to the upstream router, then the ASA drops the ARP request from the downstream router to the upstream router.

The **standby** keyword and address is used for failover.

Example

The following example sets the BVI 2 address and standby address:

```
ciscoasa(config)# interface bvi 2
ciscoasa(config-if)# ip address 10.1.3.1 255.255.255.0 standby 10.1.3.2
```

Configure General Bridge Group Member Interface Parameters

This procedure describes how to set the name, security level, and bridge group for each bridge group member interface.

Before you begin

- The same bridge group can include different types of interfaces: physical interfaces, VLAN subinterfaces, VNI interfaces, EtherChannels, and redundant interfaces. The Management interface is not supported.
- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.
- For transparent mode, do not use this procedure for Management interfaces; see Configure a Management Interface for Transparent Mode, on page 460 to configure the Management interface.

Procedure

Step 1 Enter interface configuration mode:

interface *id*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

The interface ID can be:

- **redundant**
- **port-channel**
- *physical*—For example, **ethernet**, **gigabitethernet**, **tengigabitethernet**. Management interfaces are not supported. Refer to the hardware installation guide for your model for interface names.
- *physical_or_port-channel_or_redundant***.***subinterface*—For example, **gigabitethernet0/0.100**, **port-channel1.100**. or **redundant2.100**.

- **vni**
- **vlan**
- *mapped_name*—For multiple context mode.

Step 2 Assign the interface to a bridge group:

bridge-group *number*

Example:

```
ciscoasa(config-if)# bridge-group 1
```

The *number* is an integer between 1 and 250, and must match the BVI interface number. You can assign up to 4 interfaces to a bridge group. You cannot assign the same interface to more than one bridge group.

Step 3 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif inside1
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 4 Set the security level:

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 50
```

The *number* is an integer between 0 (lowest) and 100 (highest)..

Related Topics

Configure the MTU and TCP MSS, on page 482

Configure a Management Interface for Transparent Mode

In transparent firewall mode, all interfaces must belong to a bridge group. The only exception is the Management interface (either the physical interface, a subinterface (if supported for your model), or an EtherChannel interface comprised of Management interfaces (if you have multiple Management interfaces)) which you can configure as a separate management interface; for the Firepower 9300 chassis, the management interface ID depends on the mgmt-type interface that you assigned to the ASA logical device. You cannot use any other interface types as management interfaces. You can configure one management interface in single mode or per context. For more information see Management Interface for Transparent Mode, on page 403.

Before you begin

- Do not assign this interface to a bridge group; a non-configurable bridge group (ID 301) is automatically added to your configuration. This bridge group is not included in the bridge group limit.
- If your model does not include a Management interface, you must manage the transparent firewall from a data interface; skip this procedure. (For example, on the ASASM.) For the Firepower 9300 chassis, the management interface ID depends on the mgmt-type interface that you assigned to the ASA logical device.
- In multiple context mode, you cannot share any interfaces, including the Management interface, across contexts. You must connect to a data interface.
- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Enter interface configuration mode:

interface {{**port-channel** *number* | **management** *slot*/*port* | *mgmt-type_interface_id* }[**.** *subinterface*] | *mapped_name*}

Example:

```
ciscoasa(config)# interface management 0/0.1
```

The **port-channel** *number* argument is the EtherChannel interface ID, such as **port-channel 1**. The EtherChannel interface must have only Management member interfaces.

Redundant interfaces do not support Management *slot*/*port* interfaces as members. You can, however, set a redundant interface comprised of non-Management interfaces as management-only.

In multiple context mode, enter the *mapped_name* if one was assigned using the **allocate-interface** command.

For the Firepower 9300 chassis, specify the interface ID for the mgmt type interface (individual or EtherChannel) that you assigned to the ASA logical device.

Step 2 Name the interface:

nameif *name*

Example:

```
ciscoasa(config-if)# nameif management
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 3 Set the IP address using one of the following methods.

- Set the IP address manually:

  For use with failover, you must set the IP address and standby address manually; DHCP is not supported.

The *ip_address* and *mask* arguments set the interface IP address and subnet mask.

The standby *ip_address* argument is used for failover.

ip address *ip_address* [*mask*] [**standby** *ip_address*]

Example:

```
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
```

- Obtain an IP address from a DHCP server:

  ip address dhcp [**setroute**]

  Example:

  ```
  ciscoasa(config-if)# ip address dhcp
  ```

  The **setroute** keyword lets the ASA use the default route supplied by the DHCP server.

  Reenter this command to reset the DHCP lease and request a new lease.

  If you do not enable the interface using the no shutdown command before you enter the **ip address dhcp** command, some DHCP requests might not be sent.

Step 4 Set the security level:

security-level *number*

Example:

```
ciscoasa(config-if)# security-level 100
```

The *number* is an integer between 0 (lowest) and 100 (highest).

Configure IPv6 Addressing

This section describes how to configure IPv6 addressing.

About IPv6

This section includes information about IPv6.

IPv6 Addressing

You can configure two types of unicast addresses for IPv6:

- Global—The global address is a public address that you can use on the public network. For a bridge group, this address needs to be configured for the BVI, and not per member interface. You can also configure a global IPv6 address for the management interface in transparent mode.

- Link-local—The link-local address is a private address that you can only use on the directly-connected network. Routers do not forward packets using link-local addresses; they are only for communication on a particular physical network segment. They can be used for address configuration or for the Neighbor Discovery functions such as address resolution. In a bridge group, only member interfaces have link-local addresses; the BVI does not have a link-local address.

At a minimum, you need to configure a link-local address for IPv6 to operate. If you configure a global address, a link-local address is automatically configured on the interface, so you do not also need to specifically configure a link-local address. For bridge group member interfaces, when you configure the global address on the BVI, the ASA automatically generates link-local addresses for member interfaces. If you do not configure a global address, then you need to configure the link-local address, either automatically or manually.

Note If you want to only configure the link-local addresses, see the **ipv6 enable** (to auto-configure) or **ipv6 address link-local** (to manually configure) command in the command reference.

Modified EUI-64 Interface IDs

RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture requires that the interface identifier portion of all unicast IPv6 addresses, except those that start with binary value 000, be 64 bits long and be constructed in Modified EUI-64 format. The ASA can enforce this requirement for hosts attached to the local link.

When this feature is enabled on an interface, the source addresses of IPv6 packets received on that interface are verified against the source MAC addresses to ensure that the interface identifiers use the Modified EUI-64 format. If the IPv6 packets do not use the Modified EUI-64 format for the interface identifier, the packets are dropped and the following system log message is generated:

```
325003: EUI-64 source address check failed.
```

The address format verification is only performed when a flow is created. Packets from an existing flow are not checked. Additionally, the address verification can only be performed for hosts on the local link.

Configure a Global IPv6 Address

To configure a global IPv6 address for any routed mode interface and for the transparent mode BVI, perform the following steps.

Note Configuring the global address automatically configures the link-local address, so you do not need to configure it separately.

For subinterfaces, we recommend that you also set the MAC address manually, because they use the same burned-in MAC address of the parent interface. IPv6 link-local addresses are generated based on the MAC address, so assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA. See Manually Configure the MAC Address, on page 480.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Enter interface configuration mode:

interface *id*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
```

In transparent mode, specify the BVI:

Example:

```
ciscoasa(config)# interface bvi 1
```

In transparent mode, in addition to the BVI, you can also specify a Management interface:

Example:

```
ciscoasa(config)# interface management 1/1
```

Step 2 (Routed interface) Set the IP address using one of the following methods.

- Enable stateless autoconfiguration on the interface:

  ipv6 address autoconfig

  Enabling stateless autoconfiguration on the interface configures IPv6 addresses based on prefixes received in Router Advertisement messages. A link-local address, based on the Modified EUI-64 interface ID, is automatically generated for the interface when stateless autoconfiguration is enabled.

  Note Although RFC 4862 specifies that hosts configured for stateless autoconfiguration do not send Router Advertisement messages, the ASA does send Router Advertisement messages in this case. See the **ipv6 nd suppress-ra** command to suppress messages.

- Manually assign a global address to the interface:

  ipv6 address *ipv6_address*/*prefix-length* [**standby** *ipv6_address*]

  Example:

  ```
  ciscoasa(config-if)# ipv6 address 2001:0DB8:BA98::3210/64 standby 2001:0DB8:BA98::3211
  ```

  When you assign a global address, the link-local address is automatically created for the interface.

  standby specifies the interface address used by the secondary unit or failover group in a failover pair.

- Assign a global address to the interface by combining the specified prefix with an interface ID generated from the interface MAC address using the Modified EUI-64 format:

  ipv6 address *ipv6-prefix*/*prefix-length* **eui-64**

  Example:

  ```
  ciscoasa(config-if)# ipv6 address 2001:0DB8:BA98::/64 eui-64
  ```

  When you assign a global address, the link-local address is automatically created for the interface.

  You do not need to specify the standby address; the interface ID will be generated automatically.

Step 3 (BVI interface) Manually assign a global address to the BVI. For a management interface in Transparent mode, use this method as well.

ipv6 address *ipv6_address*/*prefix-length* [**standby** *ipv6_address*]

Example:

```
ciscoasa(config-if)# ipv6 address 2001:0DB8::BA98:0:3210/48
```

When you assign a global address, the link-local address is automatically created for the interface.

standby specifies the interface address used by the secondary unit or failover group in a failover pair.

Step 4 (Optional) Enforce the use of Modified EUI-64 format interface identifiers in IPv6 addresses on a local link:

ipv6 enforce-eui64 *if_name*

Example:

```
ciscoasa(config)# ipv6 enforce-eui64 inside
```

The *if_name* argument is the name of the interface, as specified by the **nameif** command, on which you are enabling the address format enforcement.

Configure IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMPv6 messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the readability of a neighbor, and keep track of neighboring routers.

Nodes (hosts) use neighbor discovery to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid. Hosts also use neighbor discovery to find neighboring routers that are willing to forward packets on their behalf. In addition, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses. When a router or the path to a router fails, a host actively searches for functioning alternates.

-

Procedure

Step 1 Specify the IPv6 interface you want to configure.

interface *name*

Example:

```
ciscoasa(config)# interface gigabitethernet 0/0
ciscoasa(config-if)#
```

Step 2 Specify the number Duplicate Address Detection (DAD) attempts.

ipv6 nd dad attempts *value*

Valid values for the *value* argument range from 0 to 600. A 0 value disables DAD processing on the specified interface. The default is 1 message.

DAD ensures the uniqueness of new unicast IPv6 addresses before they are assigned, and ensures that duplicate IPv6 addresses are detected in the network on a link basis. The ASA uses neighbor solicitation messages to perform DAD.

When a duplicate address is identified, the state of the address is set to DUPLICATE, the address is not used, and the following error message is generated:

```
325002: Duplicate address ipv6_address/MAC_address on interface
```

If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface. If the duplicate address is a global address, the address is not used.

Example:

```
ciscoasa(config-if)# ipv6 nd dad attempts 20
```

Step 3 Set the interval between IPv6 neighbor solicitation retransmissions.

ipv6 nd ns-interval *value*

Values for the *value* argument range from 1000 to 3600000 milliseconds.

Neighbor solicitation messages (ICMPv6 Type 135) are sent on the local link by nodes attempting to discover the link-layer addresses of other nodes on the local link. After receiving a neighbor solicitation message, the destination node replies by sending a neighbor advertisement message (ICPMv6 Type 136) on the local link.

After the source node receives the neighbor advertisement, the source node and destination node can communicate. Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verifying the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link.

Example:

```
ciscoasa(config-if)# ipv6 nd ns-interval 9000
```

Step 4 Set the amount of time that a remote IPv6 node is reachable.

ipv6 nd reachable-time *value*

Values for the *value* argument range from 0 to 3600000 milliseconds. When 0 is used for the value, the reachable time is sent as undetermined. It is up to the receiving devices to set and track the reachable time value.

The neighbor reachable time enables detecting unavailable neighbors. Shorter configured times enable detecting unavailable neighbors more quickly, however, shorter times consume more IPv6 network bandwidth and processing resources in all IPv6 network devices. Very short configured times are not recommended in normal IPv6 operation.

Example:

```
ciscoasa config-if)# ipv6 nd reachable-time 1700000
```

Step 5 Set the interval between IPv6 router advertisement transmissions.

ipv6 nd ra-interval [**msec**] *value*

The **msec** keyword indicates that the value provided is in milliseconds. If this keyword is not present, the value provided is in seconds. Valid values for the *value* argument range from 3 to 1800 seconds or from 500 to 1800000 milliseconds if the **msec** keyword is provided. The default is 200 seconds.

The interval value is included in all IPv6 router advertisements that are sent out of this interface.

The interval between transmissions should be less than or equal to the IPv6 router advertisement lifetime if the ASA is configured as a default router. To prevent synchronization with other IPv6 nodes, randomly adjust the actual value used to within 20 percent of the desired value.

Example:

```
ciscoasa(config-if)# ipv6 nd ra-interval 201
```

Step 6 Specify the length of time that nodes on the local link should consider the ASA as the default router on the link.

ipv6 nd ra-lifetime [**msec**] *value*

The optional **msec** keyword indicates that the value provided is in milliseconds. Otherwise, the value is in seconds. Values for the *value* argument range from 0 to 9000 seconds. Entering 0 indicates that the ASA should not be considered a default router on the selected interface.

The router lifetime value is included in all IPv6 router advertisements sent out of the interface. The value indicates the usefulness of the ASA as a default router on this interface.

Example:

```
ciscoasa(config-if)# ipv6 nd ra-lifetime 2000
```

Step 7 Suppress router advertisements.

ipv6 nd suppress-ra

Router advertisement messages (ICMPv6 Type 134) are automatically sent in response to router solicitation messages (ICMPv6 Type 133). Router solicitation messages are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message.

You may want to disable these messages on any interface for which you do not want the ASA to supply the IPv6 prefix (for example, the outside interface).

Entering this command causes the ASA to appear as a regular IPv6 neighbor on the link and not as an IPv6 router.

Step 8 Add a flag to IPv6 router advertisements to inform IPv6 autoconfiguration clients to use DHCPv6 to obtain an IPv6 address, in addition to the derived stateless autoconfiguration address.

ipv6 nd managed-config-flag

This option sets the Managed Address Config flag in the IPv6 router advertisement packet.

Step 9 Add a flag to IPv6 router advertisements to inform IPv6 autoconfiguration clients to use DHCPv6 to obtain the DNS server address, or other information.

ipv6 nd other-config-flag

This option sets the Other Address Config flag in the IPv6 router advertisement packet.

Step 10 Configure which IPv6 prefixes are included in IPv6 router advertisements:

ipv6 nd prefix {*ipv6_prefix/prefix_length* | **default**} [*valid_lifetime preferred_lifetime* | **at** *valid_date preferred_date*] [**no-advertise**] [**no-autoconfig**] [ ] [**off-link**]

The prefix advertisement can be used by neighboring devices to autoconfigure their interface addresses. Stateless autoconfiguration uses IPv6 prefixes provided in router advertisement messages to create the global unicast address from the link-local address.

By default, prefixes configured as addresses on an interface using the **ipv6 address** command are advertised in router advertisements. If you configure prefixes for advertisement using the **ipv6 nd prefix** command, then only these prefixes are advertised.

For stateless autoconfiguration to work correctly, the advertised prefix length in router advertisement messages must always be 64 bits.

- **default**—Indicates that the default prefix is used.
- *valid_lifetime preferred_lifetime* —Specifies the amount of time that the specified IPv6 prefix is advertised as being valid and preferred. An address has no restrictions during the preferred lifetime. After the preferred lifetime expires, the address goes into a deprecated state; while an address is in a deprecated state, its use is discouraged, but not strictly forbidden. After the valid lifetime expires, the address becomes invalid and cannot be used. The valid lifetime must be greater than or equal to the preferred lifetime. Values range from 0 to 4294967295 seconds. The maximum value represents infinity, which can also be specified with the **infinite** keyword. The valid lifetime default is 2592000 (30 days). The preferred lifetime default is 604800 (7 days).
- **at** *valid_date preferred_date*—Indicates a specific date and time at which the prefix expires. Specify the date as the *month_name day hh***:***mm*. For example, enter `dec 1 13:00`.
- **no-advertise**—Disables advertisement of the prefix.
- **no-autoconfig**—Specifies that the prefix cannot be used for IPv6 autoconfiguration.
- **off-link**—Configures the specified prefix as off-link. The prefix will be advertised with the L-bit clear. The prefix will not be inserted into the routing table as a Connected prefix.

When onlink is on (by default), the specified prefix is assigned to the link. Nodes sending traffic to such addresses that contain the specified prefix consider the destination to be locally reachable on the link.

Example:

```
ciscoasa(config-if)# ipv6 nd prefix 2001:DB8::/32 1000 900
```

Step 11 Configure a static entry in the IPv6 neighbor discovery cache.

ipv6 neighbor *ipv6_address if_name mac_address*

The following guidelines and limitations apply for configuring a static IPv6 neighbor:

- The **ipv6 neighbor** command is similar to the **arp** command. If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry. These entries are stored in the configuration when the copy command is used to store the configuration.
- Use the **show ipv6 neighbor** command to view static entries in the IPv6 neighbor discovery cache.
- The **clear ipv6 neighbor** command deletes all entries in the IPv6 neighbor discovery cache except static entries. The **no ipv6 neighbor** command deletes a specified static entry from the neighbor discovery cache; the command does not remove dynamic entries—entries learned from the IPv6 neighbor discovery process—from the cache. Disabling IPv6 on an interface by using the **no ipv6 enable** command deletes all IPv6 neighbor discovery cache entries configured for that interface except static entries (the state of the entry changes to INCMP [Incomplete]).
- Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.
- The **clear ipv6 neighbor** command does not remove static entries from the IPv6 neighbor discovery cache; it only clears the dynamic entries.
- The ICMP syslogs generated are caused by a regular refresh of IPv6 neighbor entries. The ASA default timer for IPv6 neighbor entry is 30 seconds, so the ASA would generate ICMPv6 neighbor discovery and response packets about every 30 seconds. If the ASA has both failover LAN and state interfaces configured with IPv6 addresses, then every 30 seconds, ICMPv6 neighbor discovery and response packets will be generated by both ASAs for both configured and link-local IPv6 addresses. In addition, each packet will generate several syslogs (ICMP connection and local-host creation or teardown), so it may appear that constant ICMP syslogs are being generated. The refresh time for IPV6 neighbor entry is configurable on the regular data interface, but not configurable on the failover interface. However, the CPU impact for this ICMP neighbor discovery traffic is minimal.

Example:

```
ciscoasa(config)# ipv6 neighbor 3001:1::45A inside 002.7D1A.9472
```

Monitoring Routed and Transparent Mode Interfaces

You can monitor interface statistics, status, PPPoE.

Note For the Firepower 9300, some statistics are not shown using the ASA commands. You must view more detailed interface statistics using FXOS commands.

- /eth-uplink/fabric# **show interface**
- /eth-uplink/fabric# **show port-channel**
- /eth-uplink/fabric/interface# **show stats**

See the FXOS troubleshooting guide for more information.

Interface Statistics and Information

- **show interface**

  Displays interface statistics.

- **show interface ip brief**

  Displays interface IP addresses and status.

- **show bridge-group**

  Displays bridge group information such as interfaces assigned, MAC addresses, and IP addresses.

PPPoE

- **show ip address** *interface_name* **pppoe**

  Displays the current PPPoE client configuration information.

- **debug pppoe** {**event** | **error** | **packet**}

  Enables debugging for the PPPoE client.

- **show vpdn session** [**l2tp** | **pppoe**] [**id** *sess_id* | **packets** | **state** | **window**]

  Views the status of PPPoE sessions.

  The following examples show information provided by this command:

```
ciscoasa# show vpdn

Tunnel id 0, 1 active sessions
     time since change 65862 secs
     Remote Internet Address 10.0.0.1
    Local Internet Address 199.99.99.3
     6 packets sent, 6 received, 84 bytes sent, 0 received
Remote Internet Address is 10.0.0.1
     Session state is SESSION_UP
       Time since event change 65865 secs, interface outside
       PPP interface id is 1
       6 packets sent, 6 received, 84 bytes sent, 0 received
ciscoasa#
ciscoasa# show vpdn session
PPPoE Session Information (Total tunnels=1 sessions=1)
```

```
Remote Internet Address is 10.0.0.1
  Session state is SESSION_UP
    Time since event change 65887 secs, interface outside
    PPP interface id is 1
    6 packets sent, 6 received, 84 bytes sent, 0 received
ciscoasa#
ciscoasa# show vpdn tunnel
PPPoE Tunnel Information (Total tunnels=1 sessions=1)
Tunnel id 0, 1 active sessions
   time since change 65901 secs
   Remote Internet Address 10.0.0.1
   Local Internet Address 199.99.99.3
   6 packets sent, 6 received, 84 bytes sent, 0 received
ciscoasa#
```

IPv6 Neighbor Discovery

To monitor IPv6 neighbor discovery parameters, enter the following command:

- **show ipv6 interface**

  This command displays the usability status of interfaces configured for IPv6, including the interface name, such as “outside,” and displays the settings for the specified interface. However, it excludes the name from the command and displays the settings for all interfaces that have IPv6 enabled on them. Output for the command shows the following:

  - The name and status of the interface.
  - The link-local and global unicast addresses.
  - The multicast groups to which the interface belongs.
  - ICMP redirect and error message settings.
  - Neighbor discovery settings.
  - The actual time when the command is set to 0.
  - The neighbor discovery reachable time that is being used.

Examples for Routed and Transparent Mode Interfaces

Transparent Mode Example with 2 Bridge Groups

The following example for transparent mode includes two bridge groups of three interfaces each, plus a management-only interface:

```
interface gigabitethernet 0/0
  nameif inside1
  security-level 100
  bridge-group 1
  no shutdown
interface gigabitethernet 0/1
```

```
  nameif outside1
  security-level 0
  bridge-group 1
  no shutdown
interface gigabitethernet 0/2
  nameif dmz1
  security-level 50
  bridge-group 1
  no shutdown
interface bvi 1
  ip address 10.1.3.1 255.255.255.0 standby 10.1.3.2

interface gigabitethernet 1/0
  nameif inside2
  security-level 100
  bridge-group 2
  no shutdown
interface gigabitethernet 1/1
  nameif outside2
  security-level 0
  bridge-group 2
  no shutdown
interface gigabitethernet 1/2
  nameif dmz2
  security-level 50
  bridge-group 2
  no shutdown
interface bvi 2
  ip address 10.3.5.8 255.255.255.0 standby 10.3.5.9

interface management 0/0
  nameif mgmt
  security-level 100
  ip address 10.2.1.1 255.255.255.0 standby 10.2.1.2
  no shutdown
```

History for Routed and Transparent Mode Interfaces

Feature Name	Platform Releases	Feature Information
IPv6 Neighbor Discovery	7.0(1)	We introduced this feature. We introduced the following commands: **ipv6 nd ns-interval**, **ipv6 nd ra-lifetime**, **ipv6 nd suppress-ra**, **ipv6 neighbor**, **ipv6 nd prefix**, **ipv6 nd dad-attempts**, **ipv6 nd reachable-time**, **ipv6 address**, **ipv6 enforce-eui64**.
IPv6 support for transparent mode	8.2(1)	IPv6 support was introduced for transparent firewall mode.

Feature Name	Platform Releases	Feature Information
Bridge groups for transparent mode	8.4(1)	If you do not want the overhead of security contexts, or want to maximize your use of security contexts, you can group interfaces together in a bridge group, and then configure multiple bridge groups, one for each network. Bridge group traffic is isolated from other bridge groups. You can configure up to eight bridge groups of four interfaces each in single mode or per context. We introduced the following commands: **interface bvi**, **show bridge-group**
Address Config Flags for IPv6 DHCP Relay	9.0(1)	We introduced the following commands: **ipv6 nd managed-config-flag**, **ipv6 nd other-config-flag**.
Transparent mode bridge group maximum increased to 250	9.3(1)	The bridge group maximum was increased from 8 to 250 bridge groups. You can configure up to 250 bridge groups in single mode or per context in multiple mode, with 4 interfaces maximum per bridge group. We modified the following commands: **interface bvi**, **bridge-group**

CHAPTER 16

Advanced Interface Configuration

This chapter describes how to configure MAC addresses for interfaces, how to set the maximum transmission unit (MTU), and set the TCP maximum segment size (TCP MSS), and how to allow same security level communication. Setting the correct MTU and maximum TCP segment size is essential for the best network performance.

About Advanced Interface Configuration

This section describes advanced interface settings.

About MAC Addresses

You can manually assign MAC addresses to override the default. For multiple context mode, you can automatically generate unique MAC addresses (for all interfaces assigned to a context).

Note You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

Default MAC Addresses

Default MAC address assignments depend on the type of interface.

- Physical interfaces—The physical interface uses the burned-in MAC address.

- Redundant interfaces—A redundant interface uses the MAC address of the first physical interface that you add. If you change the order of the member interfaces in the configuration, then the MAC address changes to match the MAC address of the interface that is now listed first. If you assign a MAC address to the redundant interface, then it is used regardless of the member interface MAC addresses.
- EtherChannels (Firepower Models)—For an EtherChannel, all interfaces that are part of the channel group share the same MAC address. This feature makes the EtherChannel transparent to network applications and users, because they only see the one logical connection; they have no knowledge of the individual links. The port-channel interface uses a unique MAC address from a pool; interface membership does not affect the MAC address.
- EtherChannels (ASA Models)—The port-channel interface uses the lowest-numbered channel group interface MAC address as the port-channel MAC address. Alternatively you can configure a MAC address for the port-channel interface. We recommend configuring a unique MAC address in case the group channel interface membership changes. If you remove the interface that was providing the port-channel MAC address, then the port-channel MAC address changes to the next lowest numbered interface, thus causing traffic disruption.
- Subinterfaces—All subinterfaces of a physical interface use the same burned-in MAC address. You might want to assign unique MAC addresses to subinterfaces. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.
- ASASM VLANs—For the ASASM, all VLANs use the same MAC address provided by the backplane.

Automatic MAC Addresses

In multiple context mode, auto-generation assigns unique MAC addresses to all interfaces assigned to a context.

If you manually assign a MAC address and also enable auto-generation, then the manually assigned MAC address is used. If you later remove the manual MAC address, the auto-generated address is used, if enabled.

In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface.

Because auto-generated addresses (when using a prefix) start with A2, you cannot start manual MAC addresses with A2 if you also want to use auto-generation.

The ASA generates the MAC address using the following format:

A2*xx.yyzz.zzzz*

Where *xx.yy* is a user-defined prefix or an autogenerated prefix based on the last two bytes of the interface MAC address, and *zz.zzzz* is an internal counter generated by the ASA. For the standby MAC address, the address is identical except that the internal counter is increased by 1.

For an example of how the prefix is used, if you set a prefix of 77, then the ASA converts 77 into the hexadecimal value 004D (*yyxx*). When used in the MAC address, the prefix is reversed (*xxyy*) to match the ASA native form:

A2**4D.00***zz.zzzz*

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03***zz.zzzz*

Note The MAC address format without a prefix is a legacy version. See the **mac-address auto** command in the command reference for more information about the legacy format.

About the MTU

The MTU specifies the maximum frame *payload* size that the ASA can transmit on a given Ethernet interface. The MTU value is the frame size *without* Ethernet headers, VLAN tagging, or other overhead. For example, when you set the MTU to 1500, the expected frame size is 1518 bytes including the headers, or 1522 when using VLAN. Do not set the MTU value higher to accommodate these headers.

For VXLAN, the entire Ethernet datagram is being encapsulated, so the new IP packet is larger and requires a larger MTU: you should set the ASA VTEP source interface MTU to be the network MTU + 54 bytes.

Path MTU Discovery

The ASA supports Path MTU Discovery (as defined in RFC 1191), which lets all devices in a network path between two hosts coordinate the MTU so they can standardize on the lowest MTU in the path.

Default MTU

The default MTU on the ASA is 1500 bytes. This value does not include the 18-22 bytes for the Ethernet header, VLAN tagging, or other overhead.

When you enable VXLAN on the VTEP source interface, if the MTU is less than 1554 bytes, then the ASA automatically raises the MTU to 1554 bytes. In this case, the entire Ethernet datagram is being encapsulated, so the new packet is larger and requires a larger MTU. In general, you should set the ASA source interface MTU to be the network MTU + 54 bytes.

MTU and Fragmentation

For IPv4, if an outgoing IP packet is larger than the specified MTU, it is fragmented into 2 or more frames. Fragments are reassembled at the destination (and sometimes at intermediate hops), and fragmentation can cause performance degradation. For IPv6, packets are typically not allowed to be fragmented at all. Therefore, your IP packets should fit within the MTU size to avoid fragmentation.

For TCP packets, the endpoints typically use their MTU to determine the TCP maximum segment size (MTU - 40, for example). If additional TCP headers are added along the way, for example for site-to-site VPN tunnels, then the TCP MSS might need to be adjusted down by the tunneling entity. See About the TCP MSS, on page 478.

For UDP or ICMP, the application should take the MTU into account to avoid fragmentation.

Note The ASA can receive frames larger than the configured MTU as long as there is room in memory.

MTU and Jumbo Frames

A larger MTU lets you send larger packets. Larger packets might be more efficient for your network. See the following guidelines:

- Matching MTUs on the traffic path—We recommend that you set the MTU on all ASA interfaces and other device interfaces along the traffic path to be the same. Matching MTUs prevents intermediate devices from fragmenting the packets.
- Accommodating jumbo frames—You can set the MTU up to 9198 bytes when you enable jumbo frames. The maximum is 9000 for the ASAv and the ASA on the Firepower 9300 chassis.

Note For the ASA 5585-X and the Firepower 9300, if you use VLAN tagging, the maximum MTU is 4-bytes smaller: 9194 for the ASA 5585-X and 8996 for the Firepower 9300.

About the TCP MSS

The TCP maximum segment size (MSS) is the size of the TCP payload *before* any TCP and IP headers are added. UDP packets are not affected. The client and the server exchange TCP MSS values during the three-way handshake when establishing the connection.

You can set the TCP MSS on the ASA for through traffic; by default, the maximum TCP MSS is set to 1380 bytes. This setting is useful when the ASA needs to add to the size of the packet for IPsec VPN encapsulation. However, for non-IPsec endpoints, you should disable the maximum TCP MSS on the ASA.

If you set a maximum TCP MSS, if either endpoint of a connection requests a TCP MSS that is larger than the value set on the ASA, then the ASA overwrites the TCP MSS in the request packet with the ASA maximum. If the host or server does not request a TCP MSS, then the ASA assumes the RFC 793-default value of 536 bytes (IPv4) or 1220 bytes (IPv6), but does not modify the packet. For example, you leave the default MTU as 1500 bytes. A host requests an MSS of 1500 minus the TCP and IP header length, which sets the MSS to 1460. If the ASA maximum TCP MSS is 1380 (the default), then the ASA changes the MSS value in the TCP request packet to 1380. The server then sends packets with 1380-byte payloads. The ASA can then add up to 120 bytes of headers to the packet and still fit in the MTU size of 1500.

You can also configure the minimum TCP MSS; if a host or server requests a very small TCP MSS, the ASA can adjust the value up. By default, the minimum TCP MSS is not enabled.

For to-the-box traffic, including for SSL VPN connections, this setting does not apply. The ASA uses the MTU to derive the TCP MSS: MTU - 40 (IPv4) or MTU - 60 (IPv6).

Default TCP MSS

By default, the maximum TCP MSS on the ASA is 1380 bytes. This default accommodates IPv4 IPsec VPN connections where the headers can equal up to 120 bytes; this value fits within the default MTU of 1500 bytes.

Suggested Maximum TCP MSS Setting

The default TCP MSS assumes the ASA acts as an IPv4 IPsec VPN endpoint and has an MTU of 1500. When the ASA acts as an IPv4 IPsec VPN endpoint, it needs to accommodate up to 120 bytes for TCP and IP headers.

If you change the MTU value, use IPv6, or do not use the ASA as an IPsec VPN endpoint, then you should change the TCP MSS setting. See the following guidelines:

- Normal traffic—Disable the TCP MSS limit and accept the value established between connection endpoints. Because connection endpoints typically derive the TCP MSS from the MTU, non-IPsec packets usually fit this TCP MSS.

- IPv4 IPsec endpoint traffic—Set the maximum TCP MSS to the MTU - 120. For example, if you use jumbo frames and set the MTU to 9000, then you need to set the TCP MSS to 8880 to take advantage of the new MTU.
- IPv6 IPsec endpoint traffic—Set the maximum TCP MSS to the MTU - 140.

Inter-Interface Communication

Allowing interfaces on the same security level to communicate with each other provides the following benefits:

- You can configure more than 101 communicating interfaces.

  If you use different levels for each interface and do not assign any interfaces to the same security level, you can configure only one interface per level (0 to 100).
- You want traffic to flow freely between all same security interfaces without ACLs.

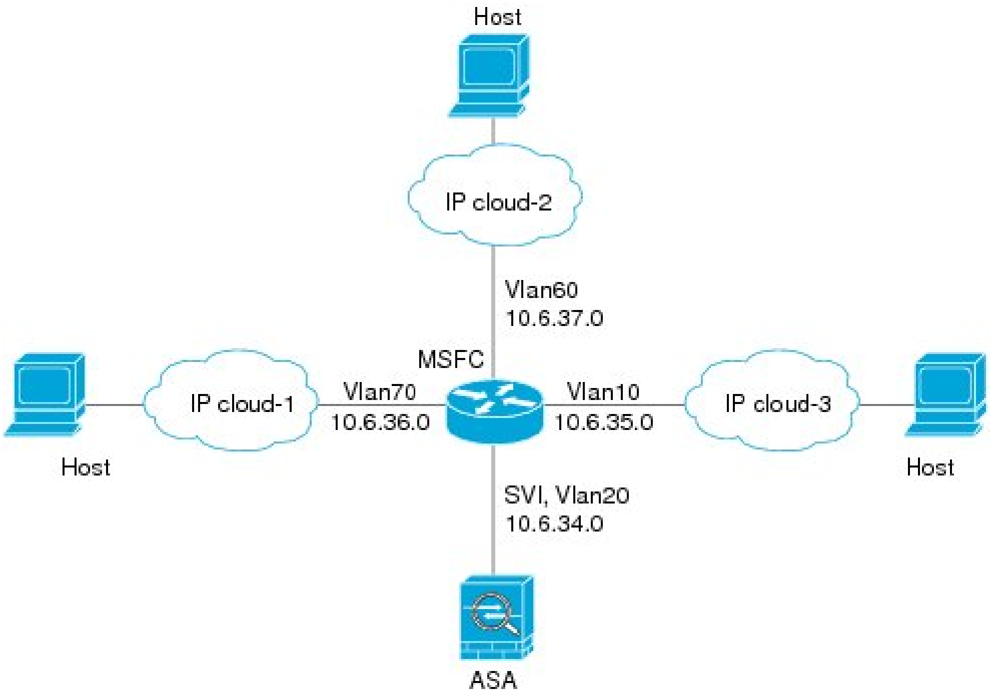
If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

Intra-Interface Communication (Routed Firewall Mode)

Intra-interface communication might be useful for VPN traffic that enters an interface, but is then routed out the same interface. The VPN traffic might be unencrypted in this case, or it might be reencrypted for another VPN connection. For example, if you have a hub and spoke VPN network, where the ASA is the hub, and remote VPN networks are spokes, for one spoke to communicate with another spoke, traffic must go into the ASA and then out again to the other spoke.

Note All traffic allowed by this feature is still subject to firewall rules. Be careful not to create an asymmetric routing situation that can cause return traffic not to traverse the ASA.

For the ASASM, before you can enable this feature, you must first correctly configure the MSFC so that packets are sent to the ASA MAC address instead of being sent directly through the switch to the destination host. The following figure shows a network where hosts on the same interface need to communicate.



The following sample configuration shows the Cisco IOS **route-map** commands used to enable policy routing in the network shown in the figure:

```
route-map intra-inter3 permit 0
  match ip address 103
  set interface Vlan20
  set ip next-hop 10.6.34.7
!
route-map intra-inter2 permit 20
  match ip address 102
  set interface Vlan20
  set ip next-hop 10.6.34.7
!
route-map intra-inter1 permit 10
  match ip address 101
  set interface Vlan20
  set ip next-hop 10.6.34.7
```

Manually Configure the MAC Address

If you need to manually assign the MAC address, you can do so using this procedure.

You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

Before you begin

In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Enter interface configuration mode:

interface *id*

Example:

```
ciscoasa(config)# interface gigabithethernet 0/0
```

Step 2 Assign a private MAC address to this interface:

mac-address *mac_address* [**standby** *mac_address*]

Example:

```
ciscoasa(config-if)# mac-address 000C.F142.4CDE
```

The *mac_address* is in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE is entered as 000C.F142.4CDE. The MAC address must not have the multicast bit set, that is, the second hexadecimal digit from the left cannot be an odd number.

The first two bytes of a manual MAC address cannot be A2 if you also want to use auto-generated MAC addresses.

For use with failover, set the **standby** MAC address. If the active unit fails over and the standby unit becomes active, the new active unit starts using the active MAC addresses to minimize network disruption, while the old active unit uses the standby address.

Automatically Assign MAC Addresses in Multiple Context Mode

This section describes how to configure auto-generation of MAC addresses. For multiple context mode, this feature assigns unique MAC addresses to all interface types that are assigned to a context.

Before you begin

- When you configure a **nameif** command for the interface, the new MAC address is generated immediately. If you enable this feature after you configure interfaces, then MAC addresses are generated for all interfaces immediately after you enable it. If you disable this feature, the MAC address for each interface reverts to the default MAC address. For example, subinterfaces of GigabitEthernet 0/1 revert to using the MAC address of GigabitEthernet 0/1.
- In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface.

- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Automatically assign private MAC addresses to each interface:

mac-address auto [**prefix** *prefix*]

If you do not enter a prefix, then the ASA autogenerates the prefix based on the last two bytes of the interface MAC address.

If you manually enter a prefix, then the *prefix* is a decimal value between 0 and 65535. This prefix is converted to a four-digit hexadecimal number, and used as part of the MAC address.

Example:

```
ciscoasa(config)# mac-address auto prefix 19
```

Configure the MTU and TCP MSS

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.
- To increase the MTU above 1500, enable jumbo frames according to Enable Jumbo Frame Support (ASA Models), on page 408. Jumbo frames are supported by default on the ASASM; you do not need to enable them.

Procedure

Step 1 Set the MTU between 300 and 9198 bytes (9000 for the ASAv and the Firepower 9300 chassis):

mtu *interface_name bytes*

Example:

```
ciscoasa(config)# mtu inside 9000
```

The default is 1500 bytes.

Note When you set the MTU for a redundant or port-channel interface, the ASA applies the setting to all member interfaces.

For many models that support jumbo frames, if you enter a value for any interface that is greater than 1500, then you need to enable jumbo frame support. See Enable Jumbo Frame Support (ASA Models), on page 408.

Note If you use VLAN tagging, the maximum value for the ASA 5585-X and Firepower 9300 chassis is reduced by 4 bytes: 9194 for the ASA 5585-X and 8996 for the Firepower 9300 chassis. Even if the ASA lets you set the MTU to a value of 9195-9198, the actual payload size will be 9194.

Step 2 Set the maximum TCP segment size in bytes, between 48 and any maximum number:

sysopt connection tcpmss [**minimum**] *bytes*

Example:

```
ciscoasa(config)# sysopt connection tcpmss 8500
ciscoasa(config)# sysopt connection tcpmss minimum 1290
```

The default value is 1380 bytes. You can disable this feature by setting bytes to **0**.

For the **minimum** keyword, sets the maximum segment size to be no less than *bytes*, between 48 and 65535. The minimum feature is disabled by default (set to 0).

Examples

The following example enables jumbo frames, increases the MTU on all interfaces, and disables the TCP MSS for non-VPN traffic (by setting the TCP MSS to 0, which means there is no limit):

```
jumbo frame-reservation
mtu inside 9198
mtu outside 9198
sysopt connection tcpmss 0
```

The following example enables jumbo frames, increases the MTU on all interfaces, and changes the TCP MSS for VPN traffic to 9078 (the MTU minus 120):

```
jumbo frame-reservation
mtu inside 9198
mtu outside 9198
sysopt connection tcpmss 9078
```

Allow Same Security Level Communication

By default, interfaces on the same security level cannot communicate with each other, and packets cannot enter and exit the same interface. This section describes how to enable inter-interface communication when interfaces are on the same security level, and how to enable intra-interface communication.

Procedure

Step 1 Enable interfaces on the same security level so that they can communicate with each other:

same-security-traffic permit inter-interface

Step 2 Enable communication between hosts connected to the same interface:

same-security-traffic permit intra-interface

History for Advanced Interface Configuration

Table 21: History for Advanced Interface Configuration

Feature Name	Releases	Feature Information
Maximum MTU is now 9198 bytes	9.1(6), 9.2(1)	The maximum MTU that the ASA can use is 9198 bytes (check for your model's exact limit at the CLI help). This value does not include the Layer 2 header. Formerly, the ASA let you specify the maximum MTU as 65535 bytes, which was inaccurate and could cause problems. If your MTU was set to a value higher than 9198, then the MTU is automatically lowered when you upgrade. In some cases, this MTU change can cause an MTU mismatch; be sure to set any connecting equipment to use the new MTU value. We modified the following command: **mtu**

CHAPTER 17

Traffic Zones

You can assign multiple interfaces to a *traffic zone*, which lets traffic from an existing flow exit or enter the ASA on any interface within the zone. This capability allows Equal-Cost Multi-Path (ECMP) routing on the ASA as well as external load balancing of traffic to the ASA across multiple interfaces.

About Traffic Zones

This section describes how you should use traffic zones in your network.

Non-Zoned Behavior

The Adaptive Security Algorithm takes into consideration the state of a packet when deciding to permit or deny the traffic. One of the enforced parameters for the flow is that traffic enters and exits the same interface. Any traffic for an existing flow that enters a different interface is dropped by the ASA.

Traffic zones let you group multiple interfaces together so that traffic entering or exiting *any* interface in the zone fulfills the Adaptive Security Algorithm security checks.

Related Topics

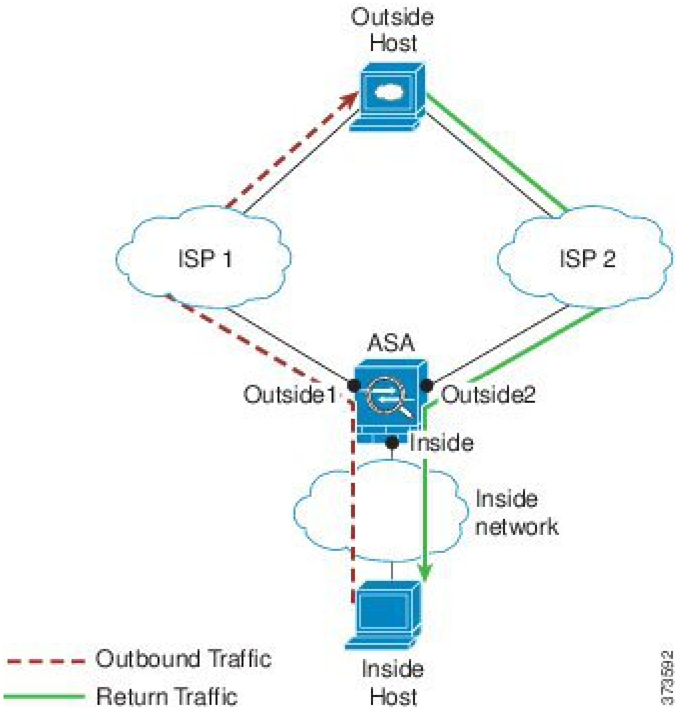
Stateful Inspection Overview, on page 15

Why Use Zones?

You can use zones to accommodate several routing scenarios.

Asymmetric Routing

In the following scenario, a connection was established between an inside host and an outside host through ISP 1 on the Outside1 interface. Due to asymmetric routing on the destination network, return traffic arrived from ISP 2 on the Outside2 interface.

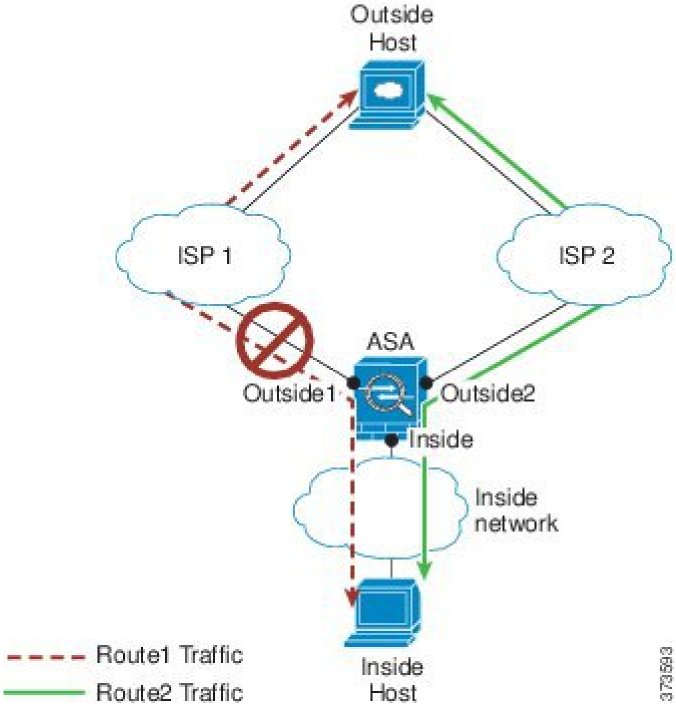


Non-Zoned Problem: The ASA maintains the connection tables on a per-interface basis. When the returning traffic arrives at Outside2, it will not match the connection table and will be dropped. For an ASA cluster, asymmetric routing when the cluster has multiple adjacencies to the same router can lead to unacceptible traffic loss.

Zoned Solution: The ASA maintains connection tables on a per-zone basis. If you group Outside1 and Outside2 into a zone, then when the returning traffic arrives at Outside2, it will match the per-zone connection table, and the connection will be allowed.

Lost Route

In the following scenario, a connection was established between an inside host and an outside host through ISP 1 on the Outside1 interface. Due to a lost or moved route between Outside1 and ISP 1, traffic needs to take a different route through ISP 2.

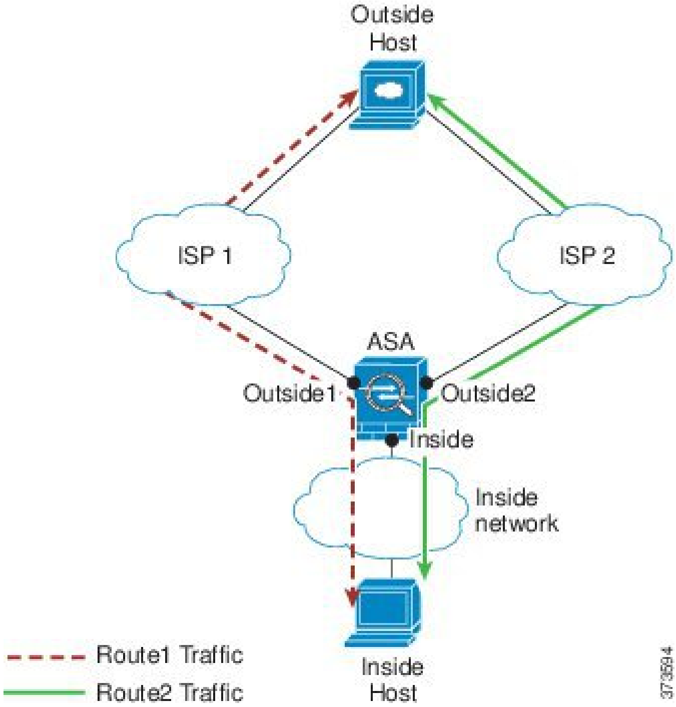


Non-Zoned Problem: The connection between the inside and outside host will be deleted; a new connection must be established using a new next-best route. For UDP, the new route will be used after a single packet drop, but for TCP, a new connection has to be reestablished.

Zoned Solution: The ASA detects the lost route and switches the flow to the new path through ISP 2. Traffic will be seamlessly forwarded without any packet drops.

Load Balancing

In the following scenario, a connection was established between an inside host and an outside host through ISP 1 on the Outside1 interface. A second connection was established through an equal cost route through ISP 2 on Outside2.



Non-Zoned Problem: Load-balancing across interfaces is not possible; you can only load-balance with equal cost routes on one interface.

Zoned Solution: The ASA load-balances connections across up to eight equal cost routes on all the interfaces in the zone.

Per-Zone Connection and Routing Tables

The ASA maintains a per-zone connection table so that traffic can arrive on any of the zone interfaces. The ASA also maintains a per-zone routing table for ECMP support.

ECMP Routing

The ASA supports Equal-Cost Multi-Path (ECMP) routing.

Non-Zoned ECMP Support

Without zones, you can have up to 8 equal cost static or dynamic routes per interface. For example, you can configure three default routes on the outside interface that specify different gateways:

```
route outside 0 0 10.1.1.2
route outside 0 0 10.1.1.3
route outside 0 0 10.1.1.4
```

In this case, traffic is load-balanced on the outside interface between 10.1.1.2, 10.1.1.3, and 10.1.1.4. Traffic is distributed among the specified gateways based on an algorithm that hashes the source and destination IP addresses.

ECMP is not supported across multiple interfaces, so you cannot define a route to the same destination on a different interface. The following route is disallowed when configured with any of the routes above:

```
route outside2 0 0 10.2.1.1
```

Zoned ECMP Support

With zones, you can have up to 8 equal cost static or dynamic routes across up to 8 interfaces within a zone. For example, you can configure three default routes across three interfaces in the zone:

```
route outside1 0 0 10.1.1.2
route outside2 0 0 10.2.1.2
route outside3 0 0 10.3.1.2
```

Similarly, your dynamic routing protocol can automatically configure equal cost routes. The ASA load-balances traffic across the interfaces with a more robust load balancing mechanism.

When a route is lost, the ASA seamlessly moves the flow to a different route.

How Connections Are Load-Balanced

The ASA load balances connections across equal cost routes using a hash made from the packet 6-tuple (source and destination IP address, source and destination port, protocol, and ingress interface). Unless the route is lost, a connection will stay on the chosen interface for its duration.

Packets within a connection are not load-balanced across routes; a connection uses a single route unless that route is lost.

The ASA does not consider the interface bandwidth or other parameters when load balancing. You should make sure all interfaces within the same zone have the same characteristics such as MTU, bandwidth, and so on.

The load-balancing algorithm is not user configurable.

Falling Back to a Route in Another Zone

When a route is lost on an interface, if there are no other routes available within the zone, then the ASA will use a route from a different interface/zone. If this backup route is used, then you may experience packet drops as with non-zoned routing support.

Interface-Based Security Policy

Zones allow traffic to and from any interface in the zone, but the security policy itself (access rules, NAT, and so on) is still applied per interface, not per zone. If you configure the same security policy for all interfaces within the zone, then you can successfully implement ECMP and load balancing for that traffic. For more information about required parallel interface configuration, see Prerequisites for Traffic Zones, on page 491.

Supported Services for Traffic Zones

The following services are supported with zones:

- Access Rules

- NAT
- Service Rules, except for QoS traffic policing.
- Routing

You can also configure to- and from-the-box services listed in To- and From-the-Box Traffic, on page 490, although full zoned support is not available.

Do not configure other services (such as VPN or Botnet Traffic Filter) for interfaces in a traffic zone; they may not function or scale as expected.

Note For detailed information about how to configure the security policy, see Prerequisites for Traffic Zones, on page 491.

Security Levels

The first interface that you add to a zone determines the security level of the zone. All additional interfaces must have the same security level. To change the security level for interfaces in a zone, you must remove all but one interface, and then change the security levels, and re-add the interfaces.

Primary and Current Interface for the Flow

Each connection flow is built based on the initial ingress and egress interfaces. These interfaces are the *primary* interfaces.

If a new egress interface is used because of route changes or asymmetric routing, then the new interfaces are the *current* interfaces.

Joining or Leaving a Zone

When you assign an interface to a zone, any connections on that interface are deleted. The connections must be reestablished.

If you remove an interface from a zone, any connections that have the interface as the primary interface are deleted. The connections must be reestablished. If the interface is the current interface, the ASA moves the connections back to the primary interface. The zone route table is also refreshed.

Intra-Zone Traffic

To allow traffic to *enter* one interface and *exit* another in the same zone, enable the **same-security permit intra-interface** command, which allows traffic to enter and exit the same interface, as well as the **same-security permit inter-interface** command, which allows traffic between same-security interfaces. Otherwise, a flow cannot be routed between two interfaces in the same zone.

To- and From-the-Box Traffic

- You cannot add management-only or management-access interfaces to a zone.

- For management traffic on regular interfaces in a zone, only asymmetric routing on existing flows is supported; there is no ECMP support.
- You can configure a management service on only one zone interface, but to take advantage of asymmetric routing support, you need to configure it on all interfaces. Even when the configurations are parallel on all interfaces, ECMP is not supported.
- The ASA supports the following to- and from-the-box services in a zone:
  - Telnet
  - SSH
  - HTTPS
  - SNMP
  - Syslog

Overlapping IP Addresses Within a Zone

For non-zoned interfaces, the ASA supports overlapping IP address networks on interfaces so long as you configure NAT properly. However, overlapping networks are not supported on interfaces in the same zone.

Prerequisites for Traffic Zones

- Configure all interface parameters including the name, IP address, and security level. Note that the security level must match for all interfaces in the zone. You should plan to group together like interfaces in terms of bandwidth and other Layer 2 properties.
- Configure the following services to match on all zone interfaces:
  - Access Rules—Apply the same access rule to all zone member interfaces, or use a global access rule.

    For example:

    ```
    access-list ZONE1 extended permit tcp any host WEBSERVER1 eq 80
    access-group ZONE1 in interface outside1
    access-group ZONE1 in interface outside2
    access-group ZONE1 in interface outside3
    ```

  - NAT—Configure the same NAT policy on all member interfaces of the zone or use a global NAT rule (in other words, use “any” to represent the zone interfaces in the NAT rule).

    Interface PAT is not supported.

    For example:

    ```
    object network WEBSERVER1
      host 10.9.9.9 255.255.255.255
      nat (inside,any) static 209.165.201.9
    ```

Note When you use interface-specific NAT and PAT pools, the ASA cannot switch connections over in case of the original interface failure.

If you use interface-specific PAT pools, multiple connections from the same host might load-balance to different interfaces and use different mapped IP addresses. Internet services that use multiple concurrent connections may not work correctly in this case.

- Service Rules—Use the global service policy, or assign the same policy to each interface in a zone.

  QoS traffic policing is not supported.

  For example:

  ```
  service-policy outside_policy interface outside1
  service-policy outside_policy interface outside2
  service-policy outside_policy interface outside3
  ```

Note For VoIP inspections, zone load balancing can cause increased out-of-order packets. This situation can occur because later packets might reach the ASA before earlier packets that take a different path. Symptoms of out-of-order packets include:

- Higher memory utilization at intermediate nodes (firewall and IDS) and the receiving end nodes if queuing is used.
- Poor video or voice quality.

To mitigate these effects, we recommend that you use IP addresses only for load distribution for VoIP traffic.

- Configure routing with ECMP zone capabilities in mind.

Guidelines for Traffic Zones

Firewall Mode

Supported in routed firewall mode only. Does not support transparent firewall mode.

Failover

- You cannot add the failover or state link to a zone.
- In Active/Active failover mode, you can assign an interface in each context to an asymmetrical routing (ASR) group. This service allows traffic returning on a similar interface on the peer unit to be restored to the original unit. You cannot configure both ASR groups and traffic zones within a context. If you configure a zone in a context, none of the context interfaces can be part of an ASR group. See Configure

Support for Asymmetrically Routed Packets (Active/Active Mode), on page 257 for more information about ASR groups.

- Only the primary interfaces for each connection are replicated to the standby unit; current interfaces are not replicated. If the standby unit becomes active, it will assign a new current interface if necessary.

Clustering

- You cannot add the cluster control link to a zone.

Additional Guidelines

- You can create a maximum of 256 zones.
- You can add the following types of interfaces to a zone:
  - Physical
  - VLAN
  - EtherChannel
  - Redundant
- You cannot add the following types of interfaces:
  - Management-only
  - Management-access
  - Failover or state link
  - Cluster control link
  - Member interfaces in an EtherChannel or redundant interface
  - VNI; also, if a regular data interface is marked as nve-only, it cannot be a member of a zone.
  - BVI, or bridge group member interfaces.
- An interface can be a member of only one zone.
- You can include up to 8 interfaces per zone.
- For ECMP, you can add up to 8 equal cost routes per zone, across all zone interfaces. You can also configure multiple routes on a single interface as part of the 8 route limit.
- When you add an interface to a zone, all static routes for those interfaces are removed.
- You cannot enable DHCP Relay on an interface in a zone.
- The ASA does not support fragmented packet reassembly for fragments that are load-balanced to separate interfaces; those fragments will be dropped.
- PIM/IGMP Multicast routing is not supported on interfaces in a zone.

Configure a Traffic Zone

Configure a named zone, and assign interfaces to the zone.

Procedure

Step 1 Add the zone:

zone *name*

Example:

```
zone outside
```

The zone name can be up to 48 characters in length.

Step 2 Add an interface to the zone:

interface *id* **zone-member** *zone_name*

Example:

```
interface gigabitethernet0/0
  zone-member outside
```

Step 3 Add more interfaces to the zone; ensure they have the same security level as the first interface you added.

Example:

```
interface gigabitethernet0/1
  zone-member outside
interface gigabitethernet0/2
  zone-member outside
interface gigabitethernet0/3
  zone-member outside
```

Examples

The following example configures an outside zone with 4 member interfaces:

```
zone outside
interface gigabitethernet0/0
  zone-member outside
interface gigabitethernet0/1
  zone-member outside
interface gigabitethernet0/2
  zone-member outside
interface gigabitethernet0/3
  zone-member outside
```

Monitoring Traffic Zones

This section describes how to monitor traffic zones.

Zone Information

- **show zone** [*name*]

  Shows zone ID, context, security level, and members.

  See the following output for the **show zone** command:

```
ciscoasa# show zone outside-zone

Zone: zone-outside id: 2
Security-level: 0
Context: test-ctx
 Zone Member(s) : 2
  outside1       GigabitEthernet0/0
  outside2       GigabitEthernet0/1
```

- **show nameif zone**

  Shows the interface names and zone names.

  See the following output for the **show nameif zone** command:

```
ciscoasa# show nameif zone
Interface                Name                   zone-name      Security
GigabitEthernet0/0       inside-1               inside-zone     100
GigabitEthernet0/1.21    inside                 inside-zone     100
GigabitEthernet0/1.31    4                                        0
GigabitEthernet0/2       outside                outside-zone   0
Management0/0            lan                                      0
```

Zone Connections

- **show conn** [**long** | **detail**] [**zone** *zone_name* [**zone** *zone_name*] [...]]

  The **show conn zone** command displays connections for a zone. The **long** and **detail** keywords show the primary interface on which the connection was built and the current interface used to forward the traffic.

  See the following output for the **show conn long zone** command:

```
ciscoasa# show conn long zone zone-inside zone zone-outside

TCP outside-zone:outside1(outside2): 10.122.122.1:1080
inside-zone:inside1(inside2): 10.121.121.1:34254, idle 0:00:02, bytes 10, flags UO
```

- **show asp table zone**

  Shows the accelerated security path tables for debugging purposes.

- **show local-host** [**zone** *zone_name* [**zone** *zone_name*] [...]]

  Shows the network states of local hosts within a zone.

  See the following output for the **show local-host zone** command. The primary interface is listed first, and the current interface is in parentheses.

```
ciscoasa# show local-host zone outside-zone

Zone:outside-zone: 4 active, 5 maximum active, 0 denied
local host: <10.122.122.1>,
    TCP flow count/limit = 3/unlimited
    TCP embryonic count to host = 0
    TCP intercept watermark = unlimited
    UDP flow count/limit = 0/unlimited

  Conn:
 TCP outside-zone:outside1(outside2): 10.122.122.1:1080
inside-zone:inside1(inside2): 10.121.121.1:34254, idle 0:00:02, bytes 10, flags UO
```

Zone Routing

- **show route zone**

  Shows the routes for zone interfaces.

  See the following output for the **show route zone** command:

```
ciscoasa# show route zone

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.105.1 255.255.255.255 [1/0] via 172.16.1.1, outside-zone:outside1
C    192.168.212.0 255.255.255.0 is directly connected, lan-zone:inside,
C    172.16.1.0 255.255.255.0 is directly connected, wan-zone:outside2
S    10.5.5.0 255.255.255.0 [1/0] via 172.16.1.1, wan-zone:outside2
O    10.2.2.1 255.255.255.255 [110/11] via 192.168.212.3, 2:09:24, lan-zone:inside
O    10.1.1.1 255.255.255.255 [110/11] via 192.168.212.2, 2:09:24, lan-zone:inside
```

- **show asp table routing**

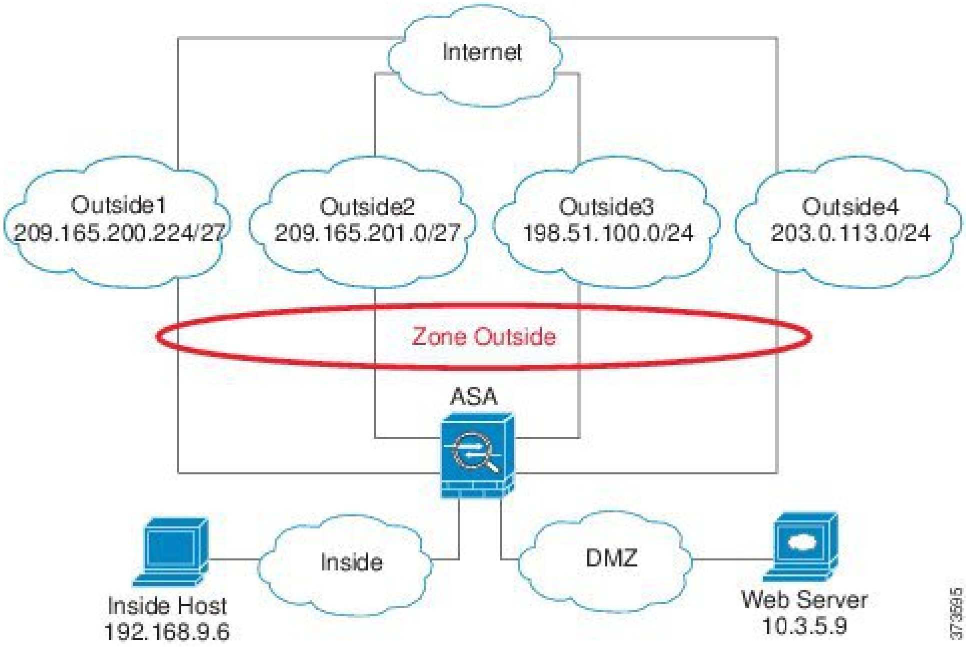
  Shows the accelerated security path tables for debugging purposes, and shows the zone associated with each route.

  See the following output for the **show asp table routing** command:

```
ciscoasa# show asp table routing
route table timestamp: 60
in   255.255.255.255 255.255.255.255 identity
in   10.1.0.1        255.255.255.255 identity
in   10.2.0.1       255.255.255.255 identity
in   10.6.6.4         255.255.255.255 identity
in   10.4.4.4         255.255.255.255 via 10.4.0.10 (unresolved, timestamp: 49)
in   172.0.0.67       255.255.255.255 identity
in   172.0.0.0       255.255.255.0   wan-zone:outside2
in   10.85.43.0      255.255.255.0   via 10.4.0.3 (unresolved, timestamp: 50)
in   10.85.45.0      255.255.255.0   via 10.4.0.20 (unresolved, timestamp: 51)
in   192.168.0.0        255.255.255.0   mgmt
in   192.168.1.0       255.255.0.0     lan-zone:inside
out  255.255.255.255 255.255.255.255 mgmt
out  172.0.0.67       255.255.255.255 mgmt
out  172.0.0.0        255.255.255.0   mgmt
out  10.4.0.0       240.0.0.0       mgmt
out  255.255.255.255 255.255.255.255 lan-zone:inside
out  10.1.0.1       255.255.255.255 lan-zone:inside
out  10.2.0.0       255.255.0.0     lan-zone:inside
out  10.4.0.0       240.0.0.0       lan-zone:inside
```

Example for Traffic Zones

The following example assigns 4 VLAN interfaces to the outside zone, and configures 4 equal cost default routes. PAT is configured for the inside interface, and a web server is available on a DMZ interface using static NAT.



```
interface gigabitethernet0/0
  no shutdown
  description outside switch 1
interface gigabitethernet0/1
  no shutdown
  description outside switch 2

interface gigabitethernet0/2
```

```
  no shutdown
  description inside switch

zone outside

interface gigabitethernet0/0.101
  vlan 101
  nameif outside1
  security-level 0
  ip address 209.165.200.225 255.255.255.224
  zone-member outside
  no shutdown

interface gigabitethernet0/0.102
  vlan 102
  nameif outside2
  security-level 0
  ip address 209.165.201.1 255.255.255.224
  zone-member outside
  no shutdown

interface gigabitethernet0/1.201
  vlan 201
  nameif outside3
  security-level 0
  ip address 198.51.100.1 255.255.255.0
  zone-member outside
  no shutdown

interface gigabitethernet0/1.202
  vlan 202
  nameif outside4
  security-level 0
  ip address 203.0.113.1 255.255.255.0
  zone-member outside
  no shutdown

interface gigabitethernet0/2.301
  vlan 301
  nameif inside
  security-level 100
  ip address 192.168.9.1 255.255.255.0
  no shutdown

interface gigabitethernet0/2.302
  vlan 302
  nameif dmz
  security-level 50
  ip address 10.3.5.1 255.255.255.0
  no shutdown

# Static NAT for DMZ web server on any destination interface
object network WEBSERVER
  host 10.3.5.9 255.255.255.255
  nat (dmz,any) static 209.165.202.129 dns

# Dynamic PAT for inside network on any destination interface
object network INSIDE
  subnet 192.168.9.0 255.255.255.0
  nat (inside,any) dynamic 209.165.202.130

# Global access rule for DMZ web server
access-list WEB-SERVER extended permit tcp any host WEBSERVER eq 80
access-group WEB-SERVER global
```

```
# 4 equal cost default routes for outside interfaces
route outside1 0 0 209.165.200.230
route outside2 0 0 209.165.201.10
route outside3 0 0 198.51.100.99
route outside4 0 0 203.0.113.87
# Static routes for NAT addresses - see redistribute static command
route dmz 209.165.202.129 255.255.255.255 10.3.5.99
route inside 209.165.202.130 255.255.255.255 192.168.9.99

# The global service policy
class-map inspection_default
  match default-inspection-traffic
policy-map type inspect dns preset_dns_map
  parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
  class inspection_default
    inspect dns preset_dns_map
    inspect ftp
    inspect h323 h225 _default_h323_map
    inspect h323 ras _default_h323_map
    inspect ip-options _default_ip_options_map
    inspect netbios
    inspect rsh
    inspect rtsp
    inspect skinny
    inspect esmtp _default_esmtp_map
    inspect sqlnet
    inspect sunrpc
    inspect tftp
    inspect sip
    inspect xdmcp
service-policy global_policy global
```

History for Traffic Zones

Feature Name	Platform Releases	Description
Traffic Zones	9.3(2)	You can group interfaces together into a traffic zone to accomplish traffic load balancing (using Equal Cost Multi-Path (ECMP) routing), route redundancy, and asymmetric routing across multiple interfaces. **Note** You cannot apply a security policy to a named zone; the security policy is interface-based. When interfaces in a zone are configured with the same access rule, NAT, and service policy, then load-balancing and asymmetric routing operate correctly. We introduced or modified the following commands: **zone**, **zone-member**, **show running-config zone**, **clear configure zone**, **show zone**, **show asp table zone**, **show nameif zone**, **show conn long**, **show local-host zone**, **show route zone**, **show asp table routing**, **clear conn zone**, **clear local-host zone**.

PART IV

Basic Settings

CHAPTER 18

Basic Settings

This chapter describes how to configure basic settings on the ASA that are typically required for a functioning configuration.

Set the Hostname, Domain Name, and the Enable and Telnet Passwords

To set the hostname, domain name, and the enable and Telnet passwords, perform the following steps.

Before you begin

Before you set the hostname, domain name, and the enable and Telnet passwords, check the following requirements:

- In multiple context mode, you can configure the hostname and domain name in both the system and context execution spaces.
- For the enable and Telnet passwords, set them in each context; they are not available in the system. When you session to the ASASM from the switch in multiple context mode, the ASASM uses the login password you set in the admin context.
- To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Specify the hostname for the ASA or for a context. The default hostname is “asa.”

hostname *name*

Example:

```
ciscoasa(config)# hostname myhostnamexample12345
```

This name can be up to 63 characters. The hostname must start and end with a letter or digit, and have only letters, digits, or a hyphen.

When you set a hostname for the ASA, that name appears in the command line prompt. If you establish sessions to multiple devices, the hostname helps you keep track of where you enter commands.

For multiple context mode, the hostname that you set in the system execution space appears in the command line prompt for all contexts. The hostname that you optionally set within a context does not appear in the command line, but can be used by the **banner** command **$(hostname)** token.

Step 2 Specify the domain name for the ASA. The default domain name is default.domain.invalid.

domain-name *name*

Example:

```
ciscoasa(config)# domain-name example.com
```

The ASA appends the domain name as a suffix to unqualified names. For example, if you set the domain name to “example.com” and specify a syslog server by the unqualified name of “jupiter,” then the ASA qualifies the name to “jupiter.example.com.”

Step 3 Change the enable password. By default, the enable password is blank.

enable password *password*

Example:

```
ciscoasa(config)# enable password Pa$$w0rd
```

The enable password lets you enter privileged EXEC mode if you do not configure enable authentication. The enable password also lets you log into ASDM with a blank username if you do not configure HTTP authentication.

The *password* argument is a case-sensitive password of 3 to 32 characters long, and can be any combination of ASCII printable characters (character codes 32-126), with the exception of spaces and the question mark.

This command changes the password for the highest privilege level (15). If you configure local command authorization, you can set enable passwords for each privilege level from 0 to 15 using the following syntax:

enable password *password* **level** *number*

The **encrypted** keyword indicates that the password is encrypted (using an MD5-based hash or a PBKDF2 (Password-Based Key Derivation Function 2) hash using SHA-512). When you define a password in the **enable password** command, the ASA encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **enable password** command does not show the actual password; it shows the encrypted password followed by the **encrypted** keyword. For example, if you enter the password “test,” the **show running-config** command output would appear as something similar to the following:

```
username user1 password DLaUiAX3l78qgoB5c7iVNw== encrypted
```

The only time you would actually enter the **encrypted** keyword at the CLI is if you are cutting and pasting a configuration file for use in another ASA, and you are using the same password.

Enter the **enable password** command without a password to set the password to the default, which is blank.

Step 4 Set the login password for Telnet access. There is no default password.

The login password is used for Telnet access when you do not configure Telnet authentication. You also use this password when accessing the ASASM from the switch with the **session** command.

passwd *password* [**encrypted**]

Example:

```
ciscoasa(config)# passwd cisco12345
```

The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. If for some reason you need to copy the password to another ASA but do not know the original password, you can enter the **passwd** command with the encrypted password and the **encrypted** keyword. Normally, you only see this keyword when you enter the **show running-config passwd** command.

Set the Date and Time

Note Do not set the date and time for the ASASM or the Firepower 2100, 4100, or 9300; the ASA receives these settings from the chassis.

Set the Time Zone and Daylight Saving Dates

To set the time zone and daylight saving date range, perform the following steps.

Procedure

Step 1 Set the time zone. By default, the time zone is UTC.

- **clock timezone** *zone* [-]*hours* [*minutes*]
  - *zone*—Specifies the time zone as a string, for example, PST for Pacific Standard Time.
  - [-]*hours*—Sets the number of hours of offset from UTC. For example, PST is -8 hours.
  - *minutes*—Sets the number of minutes of offset from UTC.

Example:

```
ciscoasa(config)# clock timezone PST -8
```

Step 2 Enter one of the following commands to change the date range for daylight saving time from the default. The default recurring date range is from 2:00 a.m. on the second Sunday in March to 2:00 a.m. on the first Sunday in November.

- Set the start and end dates for daylight saving time as a specific date in a specific year. If you use this command, you need to reset the dates every year.

  clock summer-time *zone* **date** {*day month* | *month day*} *year hh***:***mm* {*day month* | *month day*} *year hh***:***mm* [*offset*]

  - *zone* —Specifies the time zone as a string, for example, PDT for Pacific Daylight Time.
  - *day* —Sets the day of the month, from 1 to 31. You can enter the day and month as **A**pril 1 or as 1 April, for example, depending on your standard date format.
  - *month* —Sets the month as a string. You can enter the day and month as April 1 or as 1 April, depending on your standard date format.
  - *year* —Sets the year using four digits, for example, 2004. The year range is 1993 to 2035.
  - *hh:mm* —Sets the hour and minutes in 24-hour time.
  - *offset* —Sets the number of minutes to change the time for daylight saving time. By default, the value is 60 minutes.

  Example:

  ```
  ciscoasa(config)# clock summer-time PDT 1 April 2010 2:00 60
  ```

- Specify the start and end dates for daylight saving time, in the form of a day and time of the month, and not a specific date in a year. This command enables you to set a recurring date range that you do not need to change yearly.

  clock summer-time *zone* **recurring** [*week weekday month hh***:***mm week weekday month hh***:***mm*] [*offset*]

  - *zone*—Specifies the time zone as a string, for example, PDT for Pacific Daylight Time.
  - *week* —Specifies the week of the month as an integer between 1 and 4 or as the words first or last. For example, if the day might fall in the partial fifth week, then specify last.
  - *weekday* —Specifies the day of the week: Monday, Tuesday, Wednesday, and so on.
  - *month* —Sets the month as a string.
  - *hh:mm* —Sets the hour and minutes in 24-hour time.
  - *offset* —Sets the number of minutes to change the time for daylight savings time. By default, the value is 60 minutes.

  Example:

```
ciscoasa(config)# clock summer-time PDT recurring first Monday April 2:00 60
```

Set the Date and Time Using an NTP Server

NTP is used to implement a hierarchical system of servers that provide a precisely synchronized time among network systems. This kind of accuracy is required for time-sensitive operations, such as validating CRLs, which include a precise time stamp. You can configure multiple NTP servers. The ASA chooses the server with the lowest stratum—a measure of how reliable the data is.

Time derived from an NTP server overrides any time set manually.

Before you begin

In multiple context mode, you can set the time in the system configuration only.

Procedure

Step 1 (Optional) Enable MD5 authentication with an NTP server.

a) Enable authentication.

ntp authenticate

Example:

```
ciscoasa(config)# ntp authenticate
```

When you enable NTP authentication, you must also specify a key ID in the **ntp trusted-key** command and associate that key with the server with the **ntp server key** command. Configure the actual key for the ID with the **ntp authentication-key** command. If you have multiple servers, configure a separate ID for each sever.

b) Specify an authentication key ID to be a trusted key, which is required for authentication with an NTP server.

ntp trusted-key *key_id*

Example:

```
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp trusted-key 3
ciscoasa(config)# ntp trusted-key 4
```

The *key_id* argument is a value between 1 and 4294967295. You can enter multiple trusted keys for use with multiple servers.

c) Set a key to authenticate with an NTP server.

ntp authentication-key *key_id* **md5** *key*

Example:

```
ciscoasa(config)# ntp authentication-key 1 md5 aNiceKey1
ciscoasa(config)# ntp authentication-key 2 md5 aNiceKey2
ciscoasa(config)# ntp authentication-key 3 md5 aNiceKey3
ciscoasa(config)# ntp authentication-key 4 md5 aNiceKey4
```

- *key_id*—Sets the ID that you set using the **ntp trusted-key** command.
- **md5** *key*—Sets the MD5 key as a string up to 32 characters long.

Step 2 Identify an NTP server.

ntp server*ipv4_address* [**key** *key_id*] [**source** *interface_name*] [**prefer**]

Example:

```
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
```

If you enabled NTP authentication (**ntp authenticate**), you must specify the **key** *key_id* argument using the ID that you set using the **ntp trusted-key** command.

The **source** *interface_name* keyword-argument pair identifies the outgoing interface for NTP packets if you do not want to use the default interface in the routing table. Because the system does not include any interfaces in multiple context mode, specify an interface name defined in the admin context.

The **prefer** keyword sets this NTP server as the preferred server if multiple servers have similar accuracy. NTP uses an algorithm to determine which server is the most accurate and synchronizes to that one. If servers are of similar accuracy, then the **prefer** keyword specifies which of those servers to use. However, if a server is significantly more accurate than the preferred one, the ASA uses the more accurate one. For example, the ASA uses a server of stratum 2 over a server of stratum 3 that is preferred.

You can identify multiple servers; the ASA uses the most accurate server.

Set the Date and Time Manually

To set the date and time manually, perform the following steps:

Before you begin

In multiple context mode, you can set the time in the system configuration only.

Procedure

Set the date time manually.

clock set *hh***:***mm***:***ss* {*month day* | *day month*} *year*

Example:

```
ciscoasa# clock set 20:54:00 april 1 2004
```

The *hh***:***mm***:***ss* argument sets the hour, minutes, and seconds in 24-hour time. For example, enter 20:54:00 for 8:54 pm.

The day value sets the day of the month, from 1 to 31. You can enter the day and month as april 1 or as 1 april, for example, depending on your standard date format.

The month value sets the month. Depending on your standard date format, you can enter the day and month as april 1 or as 1 april.

The year value sets the year using four digits, for example, 2004. The year range is from 1993 to 2035.

The default time zone is UTC. If you change the time zone after you enter the **clock set** command using the **clock timezone** command, the time automatically adjusts to the new time zone.

This command sets the time in the hardware chip, and does not save the time in the configuration file. This time endures reboots. Unlike the other **clock** commands, this command is a privileged EXEC command. To reset the clock, you need to set a new time with the clock set command.

Configure the Master Passphrase

The master passphrase allows you to securely store plain text passwords in encrypted format and provides a key that is used to universally encrypt or mask all passwords, without changing any functionality. Features that use the master passphrase include the following:

- OSPF
- EIGRP
- VPN load balancing
- VPN (remote access and site-to-site)
- Failover
- AAA servers
- Logging
- Shared licenses

Add or Change the Master Passphrase

To add or change the master passphrase, perform the following steps.

Before you begin

- This procedure will only be accepted in a secure session, for example by console, SSH, or ASDM via HTTPS.

- If failover is enabled but no failover shared key is set, an error message appears if you change the master passphrase, informing you that you must enter a failover shared key to protect the master passphrase changes from being sent as plain text.
- Enabling or changing password encryption in Active/Standby failover causes a **write standby**, which replicates the active configuration to the standby unit. Without this replication, the encrypted passwords on the standby unit will differ even though they use the same passphrase; configuration replication ensures that the configurations are the same. For Active/Active failover, you must manually enter **write standby**. A **write standby** can cause traffic interruption in Active/Active mode, because the configuration is cleared on the secondary unit before the new configuration is synced. You should make all contexts active on the primary ASA using the **failover active group 1** and **failover active group 2** commands, enter **write standby**, and then restore the group 2 contexts to the secondary unit using the **no failover active group 2** command.

Procedure

Step 1 Set the passphrase used for generating the encryption key. The passphrase must be between 8 and 128 characters long. All characters except a backspace and double quotes are accepted for the passphrase. If you do not enter the new passphrase in the command, you are prompted for it. To change the passphrase, you must enter the old passphrase.

key config-key password-encryption [*new_passphrase* [*old_passphrase*]]

Example:

```
ciscoasa(config)# key config-key password-encryption
Old key: bumblebee
New key: haverford
Confirm key: haverford
```

Note Use the interactive prompts to enter passwords to avoid having the passwords logged in the command history buffer.

Use the **no key config-key password-encrypt** command with caution, because it changes the encrypted passwords into plain text passwords. You may use the **no** form of this command when downgrading to a software version that does not support password encryption.

Step 2 Enable password encryption.

password encryption aes

Example:

```
ciscoasa(config)# password encryption aes
```

As soon as password encryption is enabled and the master passphrase is available, all the user passwords will be encrypted. The running configuration will show the passwords in the encrypted format.

If the passphrase is not configured at the time that password encryption is enabled, the command will succeed in anticipation that the passphrase will be available in the future.

If you later disable password encryption using the **no password encryption aes** command, all existing encrypted passwords are left unchanged, and as long as the master passphrase exists, the encrypted passwords will be decrypted, as required by the application.

Step 3 Save the runtime value of the master passphrase and the resulting configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

If you do not enter this command, passwords in startup configuration may still be visible if they were not saved with encryption previously. In addition, in multiple context mode the master passphrase is changed in the system context configuration. As a result, the passwords in all contexts will be affected. If the write memory command is not entered in the system context mode, but not in all user contexts, then the encrypted passwords in user contexts may be stale. Alternatively, use the write memory all command in the system context to save all configurations.

Examples

The following example shows that no previous key was present:

```
ciscoasa(config)# key config-key password-encryption 12345678
```

The following example shows that a key already exists:

```
ciscoasa(config)# key config-key password-encryption 23456789
Old key: 12345678
```

In the following example, you enter the command without parameters so that you will be prompted for keys. Because a key already exists, you are prompted for it.

```
ciscoasa(config)# key config-key password-encryption
Old key: 12345678
New key: 23456789
Confirm key: 23456789
```

In the following example, there is no existing key, so you are not prompted to supply it.

```
ciscoasa(config)# key config-key password-encryption
New key: 12345678
Confirm key: 12345678
```

Disable the Master Passphrase

Disabling the master passphrase reverts encrypted passwords into plain text passwords. Removing the passphrase might be useful if you downgrade to a previous software version that does not support encrypted passwords.

Before you begin

- You must know the current master passphrase to disable it. See Remove the Master Passphrase, on page 512 if you do not know the passphrase.
- This procedure works only in a secure session; that is, by Telnet, SSH, or ASDM via HTTPS.

  To disable the master passphrase, perform the following steps:

Procedure

Step 1 Remove the master passphrase. If you do not enter the passphrase in the command, you are prompted for it.

no key config-key password-encryption [*old_passphrase*]]

Example:

```
ciscoasa(config)# no key config-key password-encryption

Warning! You have chosen to revert the encrypted passwords to plain text.
This operation will expose passwords in the configuration and therefore
exercise caution while viewing, storing, and copying configuration.

Old key: bumblebee
```

Step 2 Save the runtime value of the master passphrase and the resulting configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

The non-volatile memory containing the passphrase will be erased and overwritten with the 0xFF pattern.

In multiple mode, the master passphrase is changed in the system context configuration. As a result, the passwords in all contexts will be affected. If the write memory command is entered in the system context mode, but not in all user contexts, then the encrypted passwords in user contexts may be stale. Alternatively, use the write memory all command in the system context to save all configurations.

Remove the Master Passphrase

You cannot recover the master passphrase. If the master passphrase is lost or unknown, you can remove it.

To remove the master passphrase, perform the following steps:

Procedure

Step 1 Remove the master key and the configuration that includes the encrypted passwords.

write erase

Example:

```
ciscoasa(config)# write erase
```

Step 2 Reload the ASA with the startup configuration, without any master key or encrypted passwords.

reload

Example:

```
ciscoasa(config)# reload
```

Configure the DNS Server

You need to configure DNS servers so that the ASA can resolve host names to IP addresses. You also must configure DNS servers to use fully qualified domain names (FQDN) network objects in access rules.

Some ASA features require use of a DNS server to access external servers by domain name; for example, the Botnet Traffic Filter feature requires a DNS server to access the dynamic database server and to resolve entries in the static database. Other features, such as the **ping** or **traceroute** command, let you enter a name that you want to ping or traceroute, and the ASA can resolve the name by communicating with a DNS server. Many SSL VPN and certificate commands also support names.

Note The ASA has limited support for using the DNS server, depending on the feature. For example, most commands require you to enter an IP address and can only use a name when you manually configure the **name** command to associate a name with an IP address and enable use of the names using the names command.

Before you begin

Make sure that you configure the appropriate routing and access rules for any interface on which you enable DNS domain lookup so you can reach the DNS server.

Procedure

Step 1 Enable the ASA to send DNS requests to a DNS server to perform a name lookup for supported commands.

dns domain-lookup *interface_name*

Example:

```
ciscoasa(config)# dns domain-lookup inside
```

Step 2 Specify the DNS server group that the ASA uses for outgoing requests.

dns server-group DefaultDNS

Example:

```
ciscoasa(config)# dns server-group DefaultDNS
```

Other DNS server groups can be configured for VPN tunnel groups. See the **tunnel-group** command in the command reference for more information.

Step 3 Specify one or more DNS servers. You may enter all six IP addresses in the same command, separated by spaces, or you can enter each command separately. The ASA tries each DNS server in order until it receives a response.

name-server *ip_address* [*ip_address2*] [...] [*ip_address6*]

Example:

```
ciscoasa(config-dns-server-group)# name-server 10.1.1.5 192.168.1.67 209.165.201.6 dmz
```

Step 4 Configure the domain name appended to the hostname if it is not fully-qualified:

domain-name *name*

Example:

```
ciscoasa(config-dns-server-group)# domain-name example.com
```

Step 5 (Optional.) Configure additional properties of the DNS server group.

Use the following commands to change the characteristics of the group, if the default settings are not appropriate for your network.

- **timeout** *seconds*—The number of seconds, from 1 to 30, to wait before trying the next DNS server. The default is 2 seconds. Each time the ASA retries the list of servers, this timeout doubles.
- **retries** *number*—The number of times, from 0 to 10, to retry the list of DNS servers when the ASA does not receive a response.
- **expire-entry-timer minutes** *number*—The number of minutes after a DNS entry expires (that is, the TTL has passed) that the entry is removed from the DNS lookup table. Removing an entry requires that the table be recompiled, so frequent removals can increase the processing load on the device. Because some DNS entries can have very short TTL (as short as three seconds), you can use this setting to virtually extend the TTL. The default is 1 minute (that is, the entry is removed one minute after the TTL has passed). The range is 1 to 65535 minutes. This option is used when resolving FQDN network objects only.
- **poll-timer minutes** *number*—The time, in minutes, of the polling cycle used to resolve FQDN network/host objects to IP addresses. FQDN objects are resolved only if they are used in a firewall policy. The timer determines the maximum time between resolutions; the DNS entry's time-to-live (TTL) value is also used to determine when to update to IP address resolution, so individual FQDNs might be resolved more frequently than the polling cycle. The default is 240 (four hours). The range is 1 to 65535 minutes.

Configure the Hardware Bypass (Cisco ISA 3000)

You can enable the hardware bypass so that traffic continues to flow between an interface pair during a power outage. Supported interface pairs are copper GigabitEthernet 1/1 & 1/2; and GigabitEthernet 1/3 & 1/4. When the hardware bypass is active, no firewall functions are in place, so make sure you understand the risks of allowing traffic through. See the following hardware bypass guidelines:

- This feature is only available on the Cisco ISA 3000 appliance.
- If you have a fiber Ethernet model, only the copper Ethernet pair (GigabitEthernet 1/1 & 1/2) supports hardware bypass.
- When the ISA 3000 loses power and goes into hardware bypass mode, only the supported interface pairs can communicate; when using the default configuration, inside1 <---> inside2, and outside1 <---> outside2 can no longer communicate. Any existing connections between these interfaces will be lost.
- We suggest that you disable TCP sequence randomization (as described in this procedure). If randomization is enabled (the default), then when the hardware bypass is activated, TCP sessions will need to be re-established. By default, the ISA 3000 rewrites the initial sequence number (ISN) of TCP connections passing through it to a random number. When the hardware bypass is activated, the ISA 3000 is no longer in the data path and does not translate the sequence numbers; the receiving client receives an unexpected sequence number and drops the connection. Even with TCP sequence randomization disabled, some TCP connections will have to be re-established because of the link that is temporarily down during the switchover.
- Cisco TrustSec connections on hardware bypass interfaces are dropped when hardware bypass is activated. When the ISA 3000 powers on and hardware bypass is deactivated, the connections are renegotiated.
- When the hardware bypass is deactivated, and traffic resumes going through the ISA 3000 data path, some existing TCP sessions need to be re-established because of the link that is temporarily down during the switchover.
- When hardware bypass is active, the Ethernet PHYs are disconnected, so the ASA is unable to determine the interface status. Interfaces may appear to be in a down state.

Before you begin

- You must attach the hardware bypass interfaces to access ports on the switch. Do not attach them to trunk ports.

Procedure

Step 1 Configure the hardware bypass to activate during a power failure:

hardware-bypass GigabitEthernet {**1/1-1/2** | **1/3-1/4**} [**sticky**]

Example:

```
ciscoasa(config)# hardware-bypass GigabitEthernet 1/1-1/2
ciscoasa(config)# hardware-bypass GigabitEthernet 1/3-1/4
```

The **sticky** keyword keeps the appliance in hardware bypass mode after the power comes back and the appliance boots up. In this case, you need to manually turn off the hardware bypass when you are ready; this option lets you control when the brief interruption in traffic occurs.

Step 2 Manually activate or deactivate the hardware bypass:

[**no**] **hardware-bypass manual GigabitEthernet** {**1/1-1/2** | **1/3-1/4**}

Example:

```
ciscoasa# hardware-bypass manual GigabitEthernet 1/1-1/2
ciscoasa# no hardware-bypass manual GigabitEthernet 1/1-1/2
```

Step 3 (Optional) Configure the hardware bypass to remain active until after the ASA FirePOWER module boots up:

hardware-bypass boot-delay module-up sfr

You must enable hardware bypass without the **sticky** option for the boot delay to operate. Without the **hardware-bypass boot-delay** command, the hardware bypass is likely to become inactive before the ASA FirePOWER module finishes booting up. This scenario can cause traffic to be dropped if you configured the module to fail-close, for example.

Step 4 Disable TCP sequence randomization. This example shows how to disable randomization for all traffic by adding the setting to the default configuration.

policy-map global_policy

class sfrclass

set connection random-sequence-number disable

If you later decide to turn it back on, replace "disable" with **enable**.

Step 5 Save the configuration.

write memory

The behavior of hardware bypass after the system comes online is determined by the configuration setting in the startup configuration, so you must save your running configuration.

Adjust ASP (Accelerated Security Path) Performance and Behavior

The ASP is an implementation layer that puts your policies and configurations into action. It is not of direct interest except during troubleshooting with the Cisco Technical Assistance Center. However, there are a few behaviors related to performance and reliability that you can adjust.

Choose a Rule Engine Transactional Commit Model

By default, when you change a rule-based policy (such as access rules), the changes become effective immediately. However, this immediacy comes with a slight cost in performance. The performance cost is more noticeable for very large rule lists in a high connections-per-second environment, for example, when you change a policy with 25,000 rules while the ASA is handling 18,000 connections per second.

The performance is affected because the rule engine compiles rules to enable faster rule lookup. By default, the system also searches uncompiled rules when evaluating a connection attempt so that new rules can be applied; because the rules are not compiled, the search takes longer.

You can change this behavior so that the rule engine uses a transactional model when implementing rule changes, continuing to use the old rules until the new rules are compiled and ready for use. With the transactional model, performance should not drop during the rule compilation. The following table clarifies the behavioral difference.

Model	Before Compilation	During Compilation	After Compilation
Default	Matches old rules.	Match new rules. (The rate for connections per second decreases.)	Matches new rules.
Transactional	Matches old rules.	Match old rules. (The rate for connections per second is unaffected.)	Matches new rules.

An additional benefit of the transactional model is that, when replacing an ACL on an interface, there is no gap between deleting the old ACL and applying the new one. This feature reduces the chances that acceptable connections may be dropped during the operation.

Tip If you enable the transactional model for a rule type, syslogs to mark the beginning and the end of the compilation are generated. These syslogs are numbered 780001 through 780004.

Use the following procedure to enable the transactional commit model for the rule engine.

Procedure

Enable the transactional commit model for the rule engine:

asp rule-engine transactional-commit *option*

Where the options are:

- **access-group**—Access rules applied globally or to interfaces.
- **nat**—Network Address Translation rules.

Example:

```
ciscoasa(config)# asp rule-engine transactional-commit access-group
```

Enable ASP Load Balancing

The ASP load balancing mechanism helps avoid the following issues:

- Overruns caused by sporadic traffic spikes on flows
- Overruns caused by bulk flows oversubscribing specific interface receive rings
- Overruns caused by relatively heavily overloaded interface receive rings, in which a single core cannot sustain the load.

ASP load balancing allows multiple cores to work simultaneously on packets that were received from a single interface receive ring. If the system drops packets, and the **show cpu** command output is far less than 100%, then this feature may help your throughput if the packets belong to many unrelated connections.

Procedure

Step 1 Enable the automatic switching on and off of ASP load balancing:

asp load-balance per-packet auto

The **auto** keyword is not available on the ASAv; you must manually enable or disable ASP load balancing.

Step 2 Manually enable ASP load balancing:

asp load-balance per-packet

ASP load balancing is enabled until you manually disable it, even if you also have the **auto** command enabled.

Step 3 Manually disable ASP load balancing:

no asp load-balance per-packet

This command only applies if you manually enabled ASP load blancing. If you also enabled the **auto** command, then the system reverts to automatically enabling or disabling ASP load balancing.

Monitoring the DNS Cache

The ASA provides a local cache of DNS information from external DNS queries that are sent for certain clientless SSL VPN and certificate commands. Each DNS translation request is first looked for in the local cache. If the local cache has the information, the resulting IP address is returned. If the local cache can not resolve the request, a DNS query is sent to the various DNS servers that have been configured. If an external DNS server resolves the request, the resulting IP address is stored in the local cache with its corresponding hostname.

See the following command for monitoring the DNS cache:

- **show dns-hosts**

  This command shows the DNS cache, which includes dynamically learned entries from a DNS server as well as manually entered name and IP addresses using the name command.

History for Basic Settings

Feature Name	Platform Releases	Description
ISA 3000 hardware bypass	9.4(1.225)	The ISA 3000 supports a hardware bypass function to allow traffic to continue flowing through the appliance when there is a loss of power. We introduced the following commands: **hardware-bypass, hardware-bypass manual, hardware-bypass boot-delay, show hardware-bypass**
Automatic ASP Load Balancing	9.3(2)	You can now enable automatic switching on and off of the ASP load balancing feature. **Note** The automatic feature is not supported on the ASAv; only manual enabling and disabling is supported. We introduced the following command: **asp load-balance per-packet auto**.
Removal of the default Telnet password	9.0(2)9.1(2)	To improve security for management access to the ASA, the default login password for Telnet was removed; you must manually set the password before you can log in using Telnet. **Note** The login password is only used for Telnet if you do not configure Telnet user authentication (the **aaa authentication telnet console** command). Previously, when you cleared the password, the ASA restored the default of "cisco." Now when you clear the password, the password is removed. The login password is also used for Telnet sessions from the switch to the ASASM (see the **session** command). For initial ASASM access, you must use the **service-module session** command, until you set a login password. We modified the following command: **password**
Password Encryption Visibility	8.4(1)	We modified the **show password encryption** command.
Master Passphrase	8.3(1)	We introduced this feature. The master passphrase allows you to securely store plain text passwords in encrypted format and provides a key that is used to universally encrypt or mask all passwords, without changing any functionality. We introduced the following commands: **key config-key password-encryption**, **password encryption aes**, **clear configure password encryption aes**, **show running-config password encryption aes, show password encryption**

CHAPTER 19

DHCP and DDNS Services

This chapter describes how to configure the DHCP server or DHCP relay as well as dynamic DNS (DDNS) update methods.

About DHCP and DDNS Services

The following topics describe the DHCP server, DHCP relay agent, and DDNS update.

About the DHCPv4 Server

DHCP provides network configuration parameters, such as IP addresses, to DHCP clients. The ASA can provide a DHCP server to DHCP clients attached to ASA interfaces. The DHCP server provides network configuration parameters directly to DHCP clients.

An IPv4 DHCP client uses a broadcast rather than a multicast address to reach the server. The DHCP client listens for messages on UDP port 68; the DHCP server listens for messages on UDP port 67.

The DHCP server for IPv6 is not supported; you can, however, enable DHCP relay for IPv6 traffic.

DHCP Options

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. The configuration parameters are carried in tagged items that are stored in the Options field of the DHCP message and the data are also called options. Vendor information is also stored in Options, and all of the vendor information extensions can be used as DHCP options.

For example, Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address preconfigured, it sends a request with option 150 or 66 to the DHCP server to obtain this information.

- DHCP option 150 provides the IP addresses of a list of TFTP servers.

- DHCP option 66 gives the IP address or the hostname of a single TFTP server.
- DHCP option 3 sets the default route.

A single request might include both options 150 and 66. In this case, the ASA DHCP server provides values for both options in the response if they are already configured on the ASA.

You can use advanced DHCP options to provide DNS, WINS, and domain name parameters to DHCP clients; DHCP option 15 is used for the DNS domain suffix.You can also use the DHCP automatic configuration setting to obtain these values or define them manually. When you use more than one method to define this information, it is passed to DHCP clients in the following sequence:

1. Manually configured settings.
2. Advanced DHCP options settings.
3. DHCP automatic configuration settings.

For example, you can manually define the domain name that you want the DHCP clients to receive and then enable DHCP automatic configuration. Although DHCP automatic configuration discovers the domain together with the DNS and WINS servers, the manually defined domain name is passed to DHCP clients with the discovered DNS and WINS server names, because the domain name discovered by the DHCP automatic configuration process is superseded by the manually defined domain name.

About the DHCP Relay Agent

You can configure a DHCP relay agent to forward DHCP requests received on an interface to one or more DHCP servers. DHCP clients use UDP broadcasts to send their initial DHCPDISCOVER messages because they do not have information about the network to which they are attached. If the client is on a network segment that does not include a server, UDP broadcasts normally are not forwarded by the ASA because it does not forward broadcast traffic. The DHCP relay agent lets you configure the interface of the ASA that is receiving the broadcasts to forward DHCP requests to a DHCP server on another interface.

Guidelines for DHCP and DDNS Services

This section includes guidelines and limitations that you should check before configuring DHCP and DDNS services.

Firewall Mode

- DHCP Relay is not supported in transparent firewall mode.
- DHCP Server is supported in transparent firewall mode on a bridge group member interface.
- DDNS is not supported in transparent firewall mode.

IPv6

Does not support IPv6 for DHCP server; IPv6 for DHCP relay is supported.

DHCPv4 Server

- The maximum available DHCP pool is 256 addresses.
- You can configure only one DHCP server on each interface. Each interface can have its own pool of addresses to use. However the other DHCP settings, such as DNS servers, domain name, options, ping timeout, and WINS servers, are configured globally and used by the DHCP server on all interfaces.
- You cannot configure an interface as a DHCP client if that interface also has DHCP server enabled; you must use a static IP address.
- You cannot configure both a DHCP server and DHCP relay on the same device, even if you want to enable them on different interfaces; you can only configure one type of service.
- ASA does not support QIP DHCP servers for use with the DHCP proxy service.
- The DHCP server does not support BOOTP requests.

DHCP Relay

- You can configure a maximum of 10 DHCPv4 relay servers in single mode and per context, global and interface-specific servers combined, with a maximum of 4 servers per interface.
- You can configure a maximum of 10 DHCPv6 relay servers in single mode and per context. Interface-specific servers for IPv6 are not supported.
- You cannot configure both a DHCP server and DHCP relay on the same device, even if you want to enable them on different interfaces; you can only configure one type of service.
- DHCP relay services are not available in transparent firewall mode. You can, however, allow DHCP traffic through using an access rule. To allow DHCP requests and replies through the ASA, you need to configure two access rules, one that allows DCHP requests from the inside interface to the outside (UDP destination port 67), and one that allows the replies from the server in the other direction (UDP destination port 68).
- For IPv4, clients must be directly-connected to the ASA and cannot send requests through another relay agent or a router. For IPv6, the ASA supports packets from another relay server.
- The DHCP clients must be on different interfaces from the DHCP servers to which the ASA relays requests.
- You cannot enable DHCP Relay on an interface in a traffic zone.
- DHCP relay is not supported on Virtual Tunnel Interfaces (VTIs).

Configure the DHCP Server

This section describes how to configure a DHCP server provided by the ASA.

Procedure

Step 1 Enable the DHCPv4 Server, on page 524.

Step 2 Configure Advanced DHCPv4 Options, on page 526.

Enable the DHCPv4 Server

To enable the DHCP server on an ASA interface, perform the following steps:

Procedure

Step 1 Create a DHCP address pool for an interface. The ASA assigns a client one of the addresses from this pool to use for a given period of time. These addresses are the local, untranslated addresses for the directly connected network.

dhcpd address *ip_address_start-ip_address_end if_name*

Example:

```
ciscoasa(config)# dhcpd address 10.0.1.101-10.0.1.110 inside
```

The address pool must be on the same subnet as the ASA interface. In transparent mode, specify a bridge group member interface.

Step 2 (Optional) (Routed mode) Automatically configure DNS, WINS, and domain name values obtained from an interface running a DHCP or PPPoE client, or from a VPN server.

dhcpd auto_config *client_if_name* [[**vpnclient-wins-override**] **interface** *if_name*]

Example:

```
ciscoasa(config)# dhcpd auto_config outside interface inside
```

If you specify DNS, WINS, or domain name parameters using the following commands, then they overwrite the parameters obtained by automatic configuration.

Step 3 (Optional) Reserve a DHCP address for a client. The ASA assigns a specific address from the configured address pool to a DHCP client based on the client's MAC address.

dhcpd reserve-address *ip_address mac_address if_name*

Example:

```
ciscoasa(config)# dhcpd reserve-address 10.0.1.109 030c.f142.4cde inside
```

The reserved address must come from the configured address pool, and the address pool must be on the same subnet as the ASA interface. In transparent mode, specify a bridge group member interface.

Step 4 (Optional) Specify the IP address(es) of the DNS server(s).

dhcpd dns *dns1* [*dns2*]

Example:

```
ciscoasa(config)# dhcpd dns 209.165.201.2 209.165.202.129
```

Step 5 (Optional) Specify the IP address(es) of the WINS server(s). You may specify up to two WINS servers.

dhcpd wins *wins1* [*wins2*]

Example:

```
ciscoasa(config)# dhcpd wins 209.165.201.5
```

Step 6 (Optional) Change the lease length to be granted to the client. The lease length equals the amount of time in seconds that the client can use its allocated IP address before the lease expires. Enter a value from 0 to 1,048,575. The default value is 3600 seconds.

dhcpd lease *lease_length*

Example:

```
ciscoasa(config)# dhcpd lease 3000
```

Step 7 (Optional) Configure the domain name.

dhcpd domain *domain_name*

Example:

```
ciscoasa(config)# dhcpd domain example.com
```

Step 8 (Optional) Configure the DHCP ping timeout value for ICMP packets. To avoid address conflicts, the ASA sends two ICMP ping packets to an address before assigning that address to a DHCP client. The default is 50 milliseconds.

dhcpd ping timeout *milliseconds*

Example:

```
ciscoasa(config)# dhcpd ping timeout 20
```

Step 9 Define a default gateway that is sent to the DHCP clients. For routed mode, if you do not use the **dhcpd option 3 ip** command, then the ASA sends the DHCP server-enabled interface IP address as the default gateway. For transparent mode, you must set **dhcpd option 3 ip** if you want to set a default gateway; the ASA itself cannot act as the default gateway.

dhcpd option 3 ip *gateway_ip*

Example:

```
ciscoasa(config)# dhcpd option 3 ip 10.10.1.1
```

Step 10 Enable the DHCP daemon within the ASA to listen for DHCP client requests on the enabled interface.

dhcpd enable *interface_name*

Example:

```
ciscoasa(config)# dhcpd enable inside
```

Specify the same interface as the **dhcpd address** range.

Configure Advanced DHCPv4 Options

The ASA supports the DHCP options listed in RFC 2132, RFC 2562, and RFC 5510 to send information. All DHCP options (1 through 255) are supported except for 1, 12, 50–54, 58–59, 61, 67, and 82.

Procedure

Step 1 Configure a DHCP option that returns one or two IP addresses:

dhcpd option *code* **ip** *addr_1* [*addr_2*]

Example:

```
ciscoasa(config)# dhcpd option 150 ip 10.10.1.1
ciscoasa(config)# dhcpd option 3 ip 10.10.1.10
```

Option 150 provides the IP address or names of one or two TFTP servers for use with Cisco IP phones. Option 3 sets the default route for Cisco IP phones.

Step 2 Configure a DHCP option that returns a text string:

dhcpd option *code* **ascii** *text*

Example:

```
ciscoasa(config)# dhcpd option 66 ascii exampleserver
```

Option 66 provides the IP address or name of a TFTP server for use with Cisco IP phones.

Step 3 Configure a DHCP option that returns a hexadecimal value.

dhcpd option code hex *value*

Example:

```
ciscoasa(config)# dhcpd option 2 hex 22.0011.01.FF1111.00FF.0000.AAAA.1111.1111.1111.11
```

Note The ASA does not verify that the option type and value that you provide match the expected type and value for the option code as defined in RFC 2132. For example, you can enter the **dhcpd option 46 ascii hello** command, and the ASA accepts the configuration, although option 46 is defined in RFC 2132 to expect a single-digit, hexadecimal value. For more information about option codes and their associated types and expected values, see RFC 2132.

The following table shows the DHCP options that are not supported by the **dhcpd option** command.

Table 22: Unsupported DHCP Options

Option Code	Description
0	DHCPOPT_PAD
1	HCPOPT_SUBNET_MASK
12	DHCPOPT_HOST_NAME
50	DHCPOPT_REQUESTED_ADDRESS
51	DHCPOPT_LEASE_TIME
52	DHCPOPT_OPTION_OVERLOAD
53	DHCPOPT_MESSAGE_TYPE
54	DHCPOPT_SERVER_IDENTIFIER
58	DHCPOPT_RENEWAL_TIME
59	DHCPOPT_REBINDING_TIME
61	DHCPOPT_CLIENT_IDENTIFIER
67	DHCPOPT_BOOT_FILE_NAME
82	DHCPOPT_RELAY_INFORMATION
255	DHCPOPT_END

Configure the DHCP Relay Agent

When a DHCP request enters an interface, the DHCP servers to which the ASA relays the request depends on your configuration. You may configure the following types of servers:

- Interface-specific DHCP servers—When a DHCP request enters a particular interface, then the ASA relays the request only to the interface-specific servers.
- Global DHCP servers—When a DHCP request enters an interface that does not have interface-specific servers configured, the ASA relays the request to all global servers. If the interface has interface-specific servers, then the global servers are not used.

Configure the DHCPv4 Relay Agent

When a DHCP request enters an interface, the ASA relays the request to the DHCP server.

Proctreats

Step 1 Do one or both of the following:

- Specify a global DHCP server IP address and the interface through which it is reachable.

  dhcprelay server *ip_address if_name*

  Example:

  ```
  ciscoasa(config)# dhcprelay server 209.165.201.5 outside
  ciscoasa(config)# dhcprelay server 209.165.201.8 outside
  ciscoasa(config)# dhcprelay server 209.165.202.150 it
  ```

- Specify the interface ID connected to the DHCP client network, and the DHCP server IP address to be used for DHCP requests that enter that interface.

  ```
  interface interface_id
    dhcprelay server ip_address
  ```

  Example:

  ```
  ciscoasa(config)# interface gigabitethernet 0/0
  ciscoasa(config)# dhcprelay server 209.165.201.6
  ciscoasa(config)# dhcprelay server 209.165.201.7
  ciscoasa(config)# interface gigabitethernet 0/1
  ciscoasa(config)# dhcprelay server 209.165.202.155
  ciscoasa(config)# dhcprelay server 209.165.202.156
  ```

  Note that you do not specify the egress interface for the requests, as in the global **dhcprelay server** command; instead, the ASA uses the routing table to determine the egress interface.

Step 2 Enable the DHCP relay service on the interface connected to the DHCP clients. You can enable DHCP relay on multiple interfaces.

dhcprelay enable *interface*

Example:

```
ciscoasa(config)# dhcprelay enable inside
ciscoasa(config)# dhcprelay enable dmz
ciscoasa(config)# dhcprelay enable eng1
ciscoasa(config)# dhcprelay enable eng2
ciscoasa(config)# dhcprelay enable mktg
```

Step 3 (Optional) Set the number of seconds allowed for DHCP relay address handling.

dhcprelay timeout *seconds*

Example:

```
ciscoasa(config)# dhcprelay timeout 25
```

Step 4 (Optional) Change the first default router address in the packet sent from the DHCP server to the address of the ASA interface.

dhcprelay setroute *interface_name*

Example:

```
ciscoasa(config)# dhcprelay setroute inside
```

This action allows the client to set its default route to point to the ASA even if the DHCP server specifies a different router.

If there is no default router option in the packet, the ASA adds one containing the interface address.

Step 5 (Optional) Configure interfaces as trusted interfaces. Do one of the following:

- Specify a DHCP client interface that you want to trust:

  ```
  interface interface_id
    dhcprelay information trusted
  ```

  Example:

  ```
  ciscoasa(config)# interface gigabitethernet 0/0
  ciscoasa(config-if)# dhcprelay information trusted
  ```

  You can configure interfaces as trusted interfaces to preserve DHCP Option 82. DHCP Option 82 is used by downstream switches and routers for DHCP snooping and IP Source Guard. Normally, if the ASA DHCP relay agent receives a DHCP packet with Option 82 already set, but the giaddr field (which specifies the DHCP relay agent address that is set by the relay agent before it forwards the packet to the server) is set to 0, then the ASA will drop that packet by default. You can now preserve Option 82 and forward the packet by identifying an interface as a trusted interface.

- Configure all client interfaces as trusted:

  dhcprelay information trust-all

  Example:

  ```
  ciscoasa(config)# dhcprelay information trust-all
  ```

Configure the DHCPv6 Relay Agent

When a DHCPv6 request enters an interface, the ASA relays the request to all DHCPv6 global servers.

Procedure

Step 1 Specify the IPv6 DHCP server destination address to which client messages are forwarded.

ipv6 dhcprelay server *ipv6_address* [*interface*]

Example:

```
ciscoasa(config)# ipv6 dhcprelay server 3FFB:C00:C18:6:A8BB:CCFF:FE03:2701
```

The *ipv6-address* argument can be a link-scoped unicast, multicast, site-scoped unicast, or global IPv6 address. Unspecified, loopback, and node-local multicast addresses are not allowed as the relay destination. The optional *interface* argument specifies the egress interface for a destination. Client messages are forwarded to the destination address through the link to which the egress interface is connected. If the specified address is a link-scoped address, then you must specify the interface.

Step 2 Enable DHCPv6 relay service on an interface.

ipv6 dhcprelay enable *interface*

Example:

```
ciscoasa(config)# ipv6 dhcprelay enable inside
```

Step 3 (Optional) Specify the amount of time in seconds that is allowed for responses from the DHCPv6 server to pass to the DHCPv6 client through the relay binding for relay address handling.

ipv6 dhcprelay timeout *seconds*

Example:

```
ciscoasa(config)# ipv6 dhcprelay timeout 25
```

Valid values for the *seconds* argument range from 1 to 3600. The default is 60 seconds.

Configure Dynamic DNS

When an interface uses DHCP IP addressing, the assigned IP address can change when the DHCP lease is renewed. When the interface needs to be reachable using a fully qualified domain name (FQDN), the IP address change can cause the DNS server resource records (RRs) to become stale. Dynamic DNS (DDNS) provides a mechanism to update DNS RRs whenever the IP address or hostname changes. You can also use DDNS for static or PPPoE IP addressing.

DDNS updates the following RRs on the DNS server: the A RR includes the name-to-IP address mapping, while the PTR RR maps addresses to names.

The ASA supports the DDNS update method is defined by RFC 2136. It does not support the Web update method. With this method, the ASA and the DHCP server use DNS requests to update the DNS RRs. The ASA or DHCP server sends a DNS request to its local DNS server for information about the hostname and, based on the response, determines the main DNS server that owns the RRs. The ASA or DHCP server then sends an update request directly to the main DNS server. See the following typical scenarios.

- The ASA updates the A RR, and the DHCP server updates the PTR RR.

Typically, the ASA "owns" the A RR, while the DHCP server "owns" the PTR RR, so both entities need to request updates separately. When the IP address or hostname changes, the ASA sends a DHCP request to the DHCP server to inform it that it needs to request a PTR RR update.

- The DHCP server updates both the A and PTR RR.

  Use this scenario if the ASA does not have the authority to update the A RR. When the IP address or hostname changes, the ASA sends a DHCP request to the DHCP server to inform it that it needs to request an A and PTR RR update.

You can configure different ownership depending on your security needs and the requirements of the main DNS server. For example, for a static address, the ASA should own the updates for both records.

Note DDNS is not supported on the BVI or bridge group member interfaces.

Before you begin

- Configure a DNS server on **Configuration** > **Device Management** > **DNS** > **DNS Client**. See Configure the DNS Server, on page 513.
- Configure the device hostname and domain name on **Configuration** > **Device Setup** > **Device Name/Password**. See Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503. If you do not specify the hostname per interface, then the device hostname is used. If you do not specify an FQDN, then for static or PPPoE IP addressing, the system domain name or the DNS server domain name is appended to the hostname.

Procedure

Step 1 Configure a DDNS update method to enable DNS requests from the ASA.

You do not need to configure a DDNS update method if the DHCP server will perform all requests.

a) Create an update method.

ddns update method *name*

Example:

```
ciscoasa(config)# ddns update method ddns1
ciscoasa(DDNS-update-method)#
```

b) Specify the standard DDNS method.

ddns [**both**]

By default, the ASA updates the A RR only. Use this setting if you want the DHCP server to update the PTR RR. If you want the ASA to update both the A and PTR RR, specify **both**. Use the **both** keyword for static or PPPoE IP addressing.

Example:

```
ciscoasa(DDNS-update-method)# ddns
```

c) (Optional) Configure the update interface between DNS requests.

interval maximum *days hours minutes seconds*

By default when all values are set to 0, update requests are sent whenever the IP address or hostname changes. To send requests regularly, set the *days* (0-364), *hours*, *minutes*, and *seconds*.

Example:

```
ciscoasa(DDNS-update-method)# interval maximum 0 0 15 0
```

d) Associate this method with an interface. See Step 2, on page 532.

Step 2 Configure interface settings for DDNS, including setting the update method, DHCP client settings, and the hostname for this interface.

a) Enter interface configuration mode.

interface *id*

Example:

```
ciscoasa(config)# interface gigabitethernet1/1
ciscoasa(config-if)#
```

b) Assign an update method.

ddns update *name*

You do not need to assign a method if you want the DHCP server to perform all updates.

Example:

```
ciscoasa(config-if)# ddns update ddns1
```

c) Assign a hostname for this interface.

ddns update hostname *hostname*

If you do not set the hostname, the device hostname is used. If you do not specify an FQDN, then the system domain name or the default domain from the DNS server group is appended (for static or PPPoE IP addressing) or the domain name from the DHCP server is appended (for DHCP IP addressing).

Example:

```
ciscoasa(config-if)# ddns update hostname asa1.example.com
```

d) Determine which records you want the DHCP server to update.

dhcp client update dns [**server** {**both** | **none**}]

The ASA sends DHCP client requests to the DHCP server. Note that the DHCP server must also be configured to support DDNS. The server can be configured to honor the client requests, or it can override the client (in which case, it will reply to the client so the client does not also try to perform updates that the server is performing). Even if the client does not request DDNS updates, the DHCP server can be configured to send updates anyway.

For static or PPPoE IP addressing, these settings are ignored.

Note You can also set these values globally for all interfaces using the **dhcp-client update dns** command. The per-interface settings take precedence over the global settings.

- Default (no keywords)—Requests that the DHCP server perform the PTR RR update. This setting works in conjunction with a DDNS update method with **ddns** A Records enabled.
- **server both**—Requests that the DHCP server perform both A and PTR RR updates. This setting does not require a DDNS update method to be associated with the interface.
- **server none**—Requests the DHCP server not to perform updates. This setting works in conjunction with a DDNS update method with **ddns both** A and PTR records enabled.

Example:

```
ciscoasa(config-if)# ddns client update dns
```

Static IP Address

The following example shows how to configure the standard DDNS method for use with a static IP address. Note that you do not configure DHCP client settings for this scenario.

```
! Define the DDNS method to update both RRs:
ddns update method ddns-2
 ddns both
interface gigabitethernet1/1
 ip address 209.165.200.225
! Associate the method with the interface:
 ddns update ddns-2
 ddns update hostname asa1.example.com
```

Example:ASA Updates A RR and DHCP Server Updates PTR RR

The following example configures the ASA to update the A RR and the DHCP server to update the PTR RR.

```
! Define the DDNS method to update the A RR:
ddns update method ddns-1
 ddns
interface gigabitethernet1/1
  ip address dhcp
! Associate the method with the interface:
  ddns update ddns-1
  ddns update hostname asa
! Set the client to update the A RR, and the server to update the PTR RR:
  dhcp client update dns
```

Example: No DHCP Server Update of RRs

The following example configures the ASA to update both the A and PTR RR, while requesting the DHCP server to update no RRs.

```
! Define the DDNS method to update both RRs:
ddns update method ddns-2
  ddns both
```

```
! Associate the method with the interface:
interface gigabitethernet1/1
  ip address dhcp
  ddns update ddns-2
  ddns update hostname asa1.example.com
! Set the client to update both RRs, and the server to update none:
  dhcp client update dns server none
```

Example: DHCP Server Updates all RRs

The following example configures the DHCP client to request that the DHCP server to update both the A and PTR RRs. Because the server performs all updates, you do not need to associated an update method with the interface.

```
interface gigabitethernet1/1
  ip address dhcp
  ddns update hostname asa
! Configure the DHCP server to update both RRs:
  dhcp client update dns server both
```

Monitoring DHCP and DDNS Services

This section includes the procedures to monitor both DHCP and DDNS services.

Monitoring DHCP Services

- **show dhcpd** {**binding** [*IP_address*] | **state** | **statistics**}

  This command shows the current DHCP server client binding, state, and statistics.

- **show dhcprelay** {**state** | **statistics**}

  This command displays the DHCP relay status and statistics.

- **show ipv6 dhcprelay binding**

  This command shows the relay binding entries that were created by the relay agent.

- **show ipv6 dhcprelay statistics**

  This command shows DHCP relay agent statistics for IPv6.

Monitoring DDNS Status

See the following command for monitoring DDNS status.

- **show ddns update** {**interface** *if_name* | **method** [*name*]}

  This command shows the DDNS update status.

  The following example show details about the DDNS update method:

  ```
  ciscoasa# show ddns update method ddns1
  ```

```
Dynamic DNS Update Method: ddns1
  IETF standardized Dynamic DNS 'A' record update
```

The following example shows information about the DDNS interface:

```
ciscoasa# show ddns update interface outside

Dynamic DNS Update on outside:
  Update Method Name            Update Destination
  test                          not available
```

History for DHCP and DDNS Services

Feature Name	Platform Releases	Description
DHCPv6 monitoring	9.4(1)	You can now monitor DHCP statistics for IPv6 and DHCP bindings for IPv6.
DHCP Relay server validates the DHCP Server identifier for replies	9.2(4)/ 9.3(3)	If the ASA DHCP relay server receives a reply from an incorrect DHCP server, it now verifies that the reply is from the correct server before acting on the reply. We did not introduce or modify any commands. We did not modify any ASDM screens. We did not introduce or modify any commands.
DHCP rebind function	9.1(4)	During the DHCP rebind phase, the client now tries to rebind to other DHCP servers in the tunnel group list. Before this release, the client did not rebind to an alternate server when the DHCP lease fails to renew. We did not introduce or modify any commands.
DHCP trusted interfaces	9.1(2)	You can now configure interfaces as trusted interfaces to preserve DHCP Option 82. DHCP Option 82 is used by downstream switches and routers for DHCP snooping and IP Source Guard. Normally, if the ASA DHCP relay agent receives a DHCP packet with Option 82 already set, but the giaddr field (which specifies the DHCP relay agent address that is set by the relay agent before it forwards the packet to the server) is set to 0, then the ASA will drop that packet by default. You can now preserve Option 82 and forward the packet by identifying an interface as a trusted interface. We introduced or modified the following commands: **dhcprelay information trusted**, **dhcprelay information trust-all**, **show running-config dhcprelay**.
DHCP relay servers per interface (IPv4 only)	9.1(2)	You can now configure DHCP relay servers per-interface, so requests that enter a given interface are relayed only to servers specified for that interface. IPv6 is not supported for per-interface DHCP relay. We introduced or modified the following commands: **dhcprelay server** (interface config mode), **clear configure dhcprelay**, **show running-config dhcprelay**.

Feature Name	Platform Releases	Description
DHCP relay for IPv6 (DHCPv6)	9.0(1)	DHCP relay support for IPv6 was added. We introduced the following commands: **ipv6 dhcprelay server**, **ipv6 dhcprelay enable**, **ipv6 dhcprelay timeout**, **clear config ipv6 dhcprelay**, **ipv6 nd managed-config-flag**, **ipv6 nd other-config-flag**, **debug ipv6 dhcp**, **debug ipv6 dhcprelay**, **show ipv6 dhcprelay binding**, **clear ipv6 dhcprelay binding**, **show ipv6 dhcprelay statistics**, and **clear ipv6 dhcprelay statistics**.
DDNS	7.0(1)	We introduced this feature. We introduced the following commands: **ddns**, **ddns update**, **dhcp client update dns**, **dhcpd update dns**, **show running-config ddns,** and **show running-config dns server-group**.
DHCP	7.0(1)	The ASA can provide a DHCP server or DHCP relay services to DHCP clients attached to ASA interfaces. We introduced the following commands: **dhcp client update dns**, **dhcpd address**, **dhcpd domain**, **dhcpd enable**, **dhcpd lease**, **dhcpd option**, **dhcpd ping timeout**, **dhcpd update dns**, **dhcpd wins**, **dhcp-network-scope**, **dhcprelay enable**, **dhcprelay server**, **dhcprelay setroute**, **dhcp-server**. **show running-config dhcpd**, and **show running-config dhcprelay**.

CHAPTER 20

Digital Certificates

This chapter describes how to configure digital certificates.

About Digital Certificates

Digital certificates provide digital identification for authentication. A digital certificate includes information that identifies a device or user, such as the name, serial number, company, department, or IP address. CAs are responsible for managing certificate requests and issuing digital certificates. CAs are trusted authorities that “sign” certificates to verify their authenticity, thereby guaranteeing the identity of the device or user.

A digital certificate also includes a copy of the public key for the user or device. A CA can be a trusted third party, such as VeriSign, or a private (in-house) CA that you establish within your organization. CAs issue digital certificates in the context of a PKI, which uses public-key or private-key encryption to ensure security.

For authentication using digital certificates, at least one identity certificate and its issuing CA certificate must exist on an ASA. This configuration allows multiple identities, roots, and certificate hierarchies. The ASA evaluates third-party certificates against CRLs, also called authority revocation lists, all the way from the identity certificate up the chain of subordinate certificate authorities.

Descriptions of several different types of available digital certificates follow:

- A CA certificate is used to sign other certificates. It is self-signed and called a root certificate. A certificate that is issued by another CA certificate is called a subordinate certificate.
- CAs also issue identity certificates, which are certificates for specific systems or hosts.
- Code-signer certificates are special certificates that are used to create digital signatures to sign code, with the signed code itself revealing the certificate origin.

The local CA integrates an independent certificate authority feature on the ASA, deploys certificates, and provides secure revocation checking of issued certificates. The local CA provides a secure, configurable, in-house authority for certificate authentication with user enrollment through a website login page.

Note CA certificates and identity certificates apply to both site-to-site VPN connections and remote access VPN connections. Procedures in this document refer to remote access VPN use in the ASDM GUI.

Tip For an example of a scenario that includes certificate configuration and load balancing, see the following URL: https://supportforums.cisco.com/docs/DOC-5964.

Public Key Cryptography

Digital signatures, enabled by public key cryptography, provide a way to authenticate devices and users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other.

In simple terms, a signature is formed when data is encrypted with a private key. The signature is attached to the data and sent to the receiver. The receiver applies the public key of the sender to the data. If the signature sent with the data matches the result of applying the public key to the data, the validity of the message is established.

This process relies on the receiver having a copy of the public key of the sender and a high degree of certainty that this key belongs to the sender, not to someone pretending to be the sender.

Obtaining the public key of a sender is normally handled externally or through an operation performed at installation. For example, most web browsers are configured with the root certificates of several CAs by default. For VPN, the IKE protocol, a component of IPsec, can use digital signatures to authenticate peer devices before setting up security associations.

Certificate Scalability

Without digital certificates, you must manually configure each IPsec peer for each peer with which it communicates; as a result, each new peer that you add to a network would require a configuration change on each peer with which it needs to communicate securely.

When you use digital certificates, each peer is enrolled with a CA. When two peers try to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new peer is added to the network, you enroll that peer with a CA and none of the other peers need modification. When the new peer attempts an IPsec connection, certificates are automatically exchanged and the peer can be authenticated.

With a CA, a peer authenticates itself to the remote peer by sending a certificate to the remote peer and performing some public key cryptography. Each peer sends its unique certificate, which was issued by the CA. This process works because each certificate encapsulates the public key for the associated peer, each certificate is authenticated by the CA, and all participating peers recognize the CA as an authenticating authority. The process is called IKE with an RSA signature.

The peer can continue sending its certificate for multiple IPsec sessions, and to multiple IPsec peers, until the certificate expires. When its certificate expires, the peer administrator must obtain a new one from the CA.

CAs can also revoke certificates for peers that no longer participate in IPsec. Revoked certificates are not recognized as valid by other peers. Revoked certificates are listed in a CRL, which each peer may check before accepting a certificate from another peer.

Some CAs have an RA as part of their implementation. An RA is a server that acts as a proxy for the CA, so that CA functions can continue when the CA is unavailable.

Key Pairs

Key pairs are RSA or Elliptic Curve Signature Algorithm (ECDSA) keys, which have the following characteristics:

- RSA keys can be used for SSH or SSL.
- SCEP enrollment supports the certification of RSA keys.
- The maximum RSA key size is 4096, and the default is 2048.
- The maximum ECDSA key length is 521, and the default is 384.
- You can generate a general purpose RSA key pair, used for both signing and encryption, or you can generate separate RSA key pairs for each purpose. Separate signing and encryption keys help to reduce exposure of the keys, because SSL uses a key for encryption but not signing. However, IKE uses a key for signing but not encryption. By using separate keys for each, exposure of the keys is minimized.

Trustpoints

Trustpoints let you manage and track CAs and certificates. A trustpoint is a representation of a CA or identity pair. A trustpoint includes the identity of the CA, CA-specific configuration parameters, and an association with one, enrolled identity certificate.

After you have defined a trustpoint, you can reference it by name in commands requiring that you specify a CA. You can configure many trustpoints.

Note If the Cisco ASA has multiple trustpoints that share the same CA, only one of these trustpoints sharing the CA can be used to validate user certificates. To control which trustpoint sharing a CA is used for validation of user certificates issued by that CA, use the **support-user-cert-validation** command.

For automatic enrollment, a trustpoint must be configured with an enrollment URL, and the CA that the trustpoint represents must be available on the network and must support SCEP.

You can export and import the keypair and issued certificates associated with a trustpoint in PKCS12 format. This format is useful to manually duplicate a trustpoint configuration on a different ASA.

Certificate Enrollment

The ASA needs a CA certificate for each trustpoint and one or two certificates for itself, depending upon the configuration of the keys used by the trustpoint. If the trustpoint uses separate RSA keys for signing and encryption, the ASA needs two certificates, one for each purpose. In other key configurations, only one certificate is needed.

The ASA supports automatic enrollment with SCEP and with manual enrollment, which lets you paste a base-64-encoded certificate directly into the terminal. For site-to-site VPNs, you must enroll each ASA. For remote access VPNs, you must enroll each ASA and each remote access VPN client.

Proxy for SCEP Requests

The ASA can proxy SCEP requests between AnyConnect and a third-party CA. The CA only needs to be accessible to the ASA if it is acting as the proxy. For the ASA to provide this service, the user must authenticate using any of the methods supported by AAA before the ASA sends an enrollment request. You can also use host scan and dynamic access policies to enforce rules of eligibility to enroll.

The ASA supports this feature only with an AnyConnect SSL or IKEv2 VPN session. It supports all SCEP-compliant CAs, including Cisco IOS CS, Windows Server 2003 CA, and Windows Server 2008 CA.

Clientless (browser-based) access does not support SCEP proxy, although WebLaunch—clientless-initiated AnyConnect—does support it.

The ASA does not support polling for certificates.

The ASA supports load balancing for this feature.

Revocation Checking

When a certificate is issued, it is valid for a fixed period of time. Sometimes a CA revokes a certificate before this time period expires; for example, because of security concerns or a change of name or association. CAs periodically issue a signed list of revoked certificates. Enabling revocation checking forces the ASA to check that the CA has not revoked a certificate each time that it uses the certificate for authentication.

When you enable revocation checking, the ASA checks certificate revocation status during the PKI certificate validation process, which can use either CRL checking, OCSP, or both. OCSP is only used when the first method returns an error (for example, indicating that the server is unavailable).

With CRL checking, the ASA retrieves, parses, and caches CRLs, which provide a complete list of revoked (and unrevoked) certificates with their certificate serial numbers. The ASA evaluates certificates according to CRLs, also called authority revocation lists, from the identity certificate up the chain of subordinate certificate authorities.

OCSP offers a more scalable method of checking revocation status in that it localizes certificate status through a validation authority, which it queries for status of a specific certificate.

Supported CA Servers

The ASA supports the following CA servers:

Cisco IOS CS, ASA Local CA, and third-party X.509 compliant CA vendors including, but not limited to:

- Baltimore Technologies
- Entrust
- Digicert
- Geotrust
- GoDaddy
- iPlanet/Netscape
- Microsoft Certificate Services
- RSA Keon
- Thawte

- VeriSign

CRLs

CRLs provide the ASA with one way of determining whether a certificate that is within its valid time range has been revoked by the issuing CA. CRL configuration is part of configuration of a trustpoint.

You can configure the ASA to make CRL checks mandatory when authenticating a certificate by using the **revocation-check crl** command. You can also make the CRL check optional by using the **revocation-check crl none** command, which allows the certificate authentication to succeed when the CA is unavailable to provide updated CRL data.

The ASA can retrieve CRLs from CAs using HTTP, SCEP, or LDAP. CRLs retrieved for each trustpoint are cached for a configurable amount of time for each trustpoint.

Note Though the CRL server responds with HTTP flag "Connection: Keep-alive" to indicate a persistent connection, ASA does not request support for persistent connection. Change the settings on the CRL server to respond with "Connection: Close" when the list is sent.

When the ASA has cached a CRL for longer than the amount of time it is configured to cache CRLs, the ASA considers the CRL too old to be reliable, or “stale.” The ASA tries to retrieve a newer version of the CRL the next time that a certificate authentication requires a check of the stale CRL.

You could receive a *revocation check* failure for a user connection/certificate if you exceed the CRL size limit of 4MB. The syslog returns a message that it has too many entries to process, if the maximum number of entries per CRL is more than 65534.

The ASA caches CRLs for an amount of time determined by the following two factors:

- The number of minutes specified with the **cache-time** command. The default value is 60 minutes.
- The NextUpdate field in the CRLs retrieved, which may be absent from CRLs. You control whether the ASA requires and uses the NextUpdate field with the **enforcenextupdate** command.

The ASA uses these two factors in the following ways:

- If the NextUpdate field is not required, the ASA marks CRLs as stale after the length of time defined by the **cache-time** command.
- If the NextUpdate field is required, the ASA marks CRLs as stale at the sooner of the two times specified by the **cache-time** command and the NextUpdate field. For example, if the **cache-time** command is set to 100 minutes and the NextUpdate field specifies that the next update is 70 minutes away, the ASA marks CRLs as stale in 70 minutes.

If the ASA has insufficient memory to store all CRLs cached for a given trustpoint, it deletes the least recently used CRL to make room for a newly retrieved CRL. Large CRLs require significant computational overhead to parse them. Hence, for better performance, use many CRLs of smaller size rather than few large CRLs, or preferably, use OCSP.

The maximimum cache size per individual CRL is 4 MB and the permissible limit of CRL entries is 65534.

OCSP

OCSP provides the ASA with a way of determining whether a certificate that is within its valid time range has been revoked by the issuing CA. OCSP configuration is part of trustpoint configuration.

OCSP localizes certificate status on a validation authority (an OCSP server, also called the *responder*) which the ASA queries for the status of a specific certificate. This method provides better scalability and more up-to-date revocation status than does CRL checking, and helps organizations with large PKI installations deploy and expand secure networks.

Note The ASA allows a five-second time skew for OCSP responses.

You can configure the ASA to make OCSP checks mandatory when authenticating a certificate by using the **revocation-check ocsp** command. You can also make the OCSP check optional by using the **revocation-check ocsp none** command, which allows the certificate authentication to succeed when the validation authority is unavailable to provide updated OCSP data.

OCSP provides three ways to define the OCSP server URL. The ASA uses these servers in the following order:

1. The OCSP URL defined in a match certificate override rule by using the **match certificate** command).
2. The OCSP URL configured by using the **ocsp url** command.
3. The AIA field of the client certificate.

Note To configure a trustpoint to validate a self-signed OCSP responder certificate, you import the self-signed responder certificate into its own trustpoint as a trusted CA certificate. Then you configure the **match certificate** command in the client certificate validating trustpoint to use the trustpoint that includes the self-signed OCSP responder certificate to validate the responder certificate. Use the same procedure for configuring validating responder certificates external to the validation path of the client certificate.

The OCSP server (responder) certificate usually signs the OCSP response. After receiving the response, the ASA tries to verify the responder certificate. The CA normally sets the lifetime of the OCSP responder certificate to a relatively short period to minimize the chance of being compromised. The CA usually also includes an ocsp-no-check extension in the responder certificate, which indicates that this certificate does not need revocation status checking. However, if this extension is not present, the ASA tries to check revocation status using the same method specified in the trustpoint. If the responder certificate is not verifiable, revocation checks fail. To avoid this possibility, use the **revocation-check none** command to configure the responder certificate validating trustpoint, and use the **revocation-check ocsp** command to configure the client certificate.

The Local CA

The local CA performs the following tasks:

- Integrates basic certificate authority operation on the ASA.
- Deploys certificates.
- Provides secure revocation checking of issued certificates.

- Provides a certificate authority on the ASA for use with browser-based and client-based SSL VPN connections.
- Provides trusted digital certificates to users, without the need to rely on external certificate authorization.
- Provides a secure, in-house authority for certificate authentication and offers straightforward user enrollment by means of a website login.

Storage for Local CA Files

The ASA accesses and implements user information, issued certificates, and revocation lists using a local CA database. This database resides in local flash memory by default, or can be configured to reside on an external file system that is mounted and accessible to the ASA.

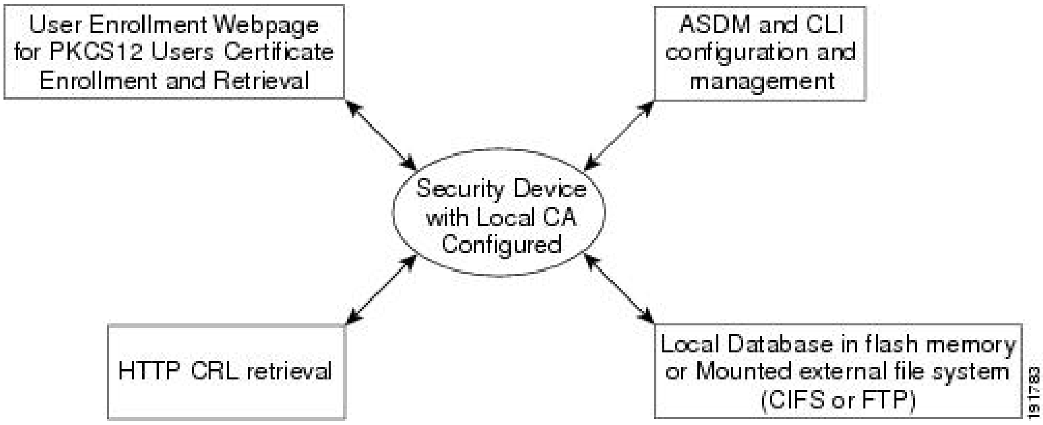
No limits exist on the number of users that can be stored in the local CA user database; however, if flash memory storage issues arise, syslogs are generated to alert the administrator to take action, and the local CA could be disabled until the storage issues are resolved. Flash memory can store a database with 3500 users or less; however, a database of more than 3500 users requires external storage.

The Local CA Server

After you configure a local CA server on the ASA, users can enroll for a certificate by logging into a website and entering a username and a one-time password that is provided by the local CA administrator to validate their eligibility for enrollment.

The following figure shows that the local CA server resides on the ASA and handles enrollment requests from website users and CRL inquiries coming from other certificate validating devices and ASAs. Local CA database and configuration files are maintained either on the ASA flash memory (default storage) or on a separate storage device.

Figure 51: The Local CA


User Enrollment Webpage for PKCS12 Users Certificate Enrollment and Retrieval
ASDM and CLI configuration and management
Security Device with Local CA Configured
HTTP CRL retrieval
Local Database in flash memory or Mounted external file system (CIFS or FTP)
191783

Certificates and User Login Credentials

The following section describes the different methods of using certificates and user login credentials (username and password) for authentication and authorization. These methods apply to IPsec, AnyConnect, and Clientless SSL VPN.

In all cases, LDAP authorization does not use the password as a credential. RADIUS authorization uses either a common password for all users or the username as a password.

User Login Credentials

The default method for authentication and authorization uses the user login credentials.

- Authentication
    - Enabled by the authentication server group setting in the tunnel group (also called ASDM Connection Profile)
    - Uses the username and password as credentials
- Authorization
    - Enabled by the authorization server group setting in the tunnel group (also called ASDM Connection Profile)
    - Uses the username as a credential

Certificates

If user digital certificates are configured, the ASA first validates the certificate. It does not, however, use any of the DNs from certificates as a username for the authentication.

If both authentication and authorization are enabled, the ASA uses the user login credentials for both user authentication and authorization.

- Authentication
    - Enabled by the authentication server group setting
    - Uses the username and password as credentials
- Authorization
    - Enabled by the authorization server group setting
    - Uses the username as a credential

If authentication is disabled and authorization is enabled, the ASA uses the primary DN field for authorization.

- Authentication
    - DISABLED (set to None) by the authentication server group setting
    - No credentials used
- Authorization
    - Enabled by the authorization server group setting
    - Uses the username value of the certificate primary DN field as a credential

Note If the primary DN field is not present in the certificate, the ASA uses the secondary DN field value as the username for the authorization request.

For example, consider a user certificate that includes the following Subject DN fields and values:

```
Cn=anyuser,OU=sales;O=XYZCorporation;L=boston;S=mass;C=us;ea=anyuser@example.com
```

If the Primary DN = EA (E-mail Address) and the Secondary DN = CN (Common Name), then the username used in the authorization request would be anyuser@example.com.

Guidelines for Digital Certificates

This section includes guidelines and limitations that you should check before configuring digital certificates.

Context Mode Guidelines

- Supported in single context mode only for third-party CAs.

Failover Guidelines

- Does not support replicating sessions in Stateful Failover.
- Does not support failover for local CAs.
- Certificates are automatically copied to the standby unit if you configure stateful failover. If you find a certificate is missing, use the **write standby** command on the active unit.

IPv6 Guidelines

Does not support IPv6.

Local CA Certificates

- Make sure that the ASA is configured correctly to support certificates. An incorrectly configured ASA can cause enrollment to fail or request a certificate that includes inaccurate information.
- Make sure that the hostname and domain name of the ASA are configured correctly. To view the currently configured hostname and domain name, enter the **show running-config** command.
- Make sure that the ASA clock is set accurately before configuring the CA. Certificates have a date and time that they become valid and expire. When the ASA enrolls with a CA and obtains a certificate, the ASA checks that the current time is within the valid range for the certificate. If it is outside that range, enrollment fails.
- Thirty days before the local CA certificate expires, a rollover replacement certificate is generated, and a syslog message informs the administrator that it is time for local CA rollover. The new local CA certificate must be imported onto all necessary devices before the current certificate expires. If the administrator does not respond by installing the rollover certificate as the new local CA certificate, validations may fail.

- The local CA certificate rolls over automatically after expiration using the same keypair. The rollover certificate is available for export in base 64 format.

  The following example shows a base 64 encoded local CA certificate:

```
MIIXlwIBAzCCF1EGCSqGSIb3DQEHAaCCF0IEghc+MIIXOjCCFzYGCSqGSIb3DQEHBqCCFycwghcjAgEAMIIXHA
YJKoZIhvcNAQcBMBsGCiqGSIb3DQEMAQMwDQQIjph4SxJoyTgCAQGAghbw3v4bFy+GGG2dJnB4OLphsUM+IG3S
DOiDwZG9n1SvtMieoxd7Hxknxbum06JDrujWKtHBIqkrm+td34qlNE1iGeP2YC94/NQ2z+4kS+uZzwcRhl1KEZ
TS1E4L0fSaC3uMTxJq2NUHYWmoc8pi4CIeLj3h7VVMy6qbx2AC8I+q57+QG5vG5l5Hi5imwtYfaWwPEdPQxaWZ
PrzoG1J8BFqdPa1jBGhAzzuSmElm3j/2dQ3Atro1G9nIsRHgV39fcBgwz4fEabHG7/Vanb+fj81d5nlOiJjDYY
bP86tvbZ2yOVZR6aKFVI0b2AfCr6PbwfC9U8Z/aF3BCyM2sN2xPJrXva94CaYrqyotZdAkSYA5KWScyEcgdqmu
BeGDKOncTknfgy0XM+fG5rb3qAXy1GkjyFI5Bm9Do6RUROoG1DSrQrKeq/hj….

END OF CERTIFICATE
```

SCEP Proxy Support

- Ensure that the ASA and the Cisco ISE Policy Service Nodes are synchronized using the same NTP server.
- AnyConnect Secure Mobility Client 3.0 or later must be running at the endpoint.
- The authentication method, configured in the connection profile for your group policy, must be set to use both AAA and certificate authentication.
- An SSL port must be open for IKEv2 VPN connections.
- The CA must be in auto-grant mode.

Local CA Certificate Database

To maintain the local CA certificate database, make sure that you save the certificate database file, LOCAL-CA-SERVER.cdb, with the **write memory** command each time that a change to the database occurs. The local CA certificate database includes the following files:

- The LOCAL-CA-SERVER.p12 file is the archive of the local CA certificate and keypair that is generated when the local CA server is initially enabled.
- The LOCAL-CA-SERVER.crl file is the actual CRL.
- The LOCAL-CA-SERVER.ser file keeps track of the issued certificate serial numbers.

Additional Guidelines

- The type of certificate you can use is constrained by the certificate types supported by the applications that will use the certificate. RSA certificates are generally supported by all applications that use certificates. But EDDSA certificates might not be supported by workstation operating systems, browsers, ASDM, or AnyConnect. For example, you need to use an RSA certificate for remote access VPN identity and authentication. For site-to-site VPN, where the ASA is the application that uses the certificate, EDDSA is supported.
- For ASAs that are configured as CA servers or clients, limit the validity period of the certificate to less than the recommended end date of 03:14:08 UTC, January 19, 2038. This guideline also applies to imported certificates from third-party vendors.

- You cannot configure the local CA when failover is enabled. You can only configure the local CA server for standalone ASAs without failover. For more information, see CSCty43366.
- When a certificate enrollment is completed, the ASA stores a PKCS12 file containing the user's keypair and certificate chain, which requires about 2 KB of flash memory or disk space per enrollment. The actual amount of disk space depends on the configured RSA key size and certificate fields. Keep this guideline in mind when adding a large number of pending certificate enrollments on an ASA with a limited amount of available flash memory, because these PKCS12 files are stored in flash memory for the duration of the configured enrollment retrieval timeout. We recommend using a key size of at least 2048.
- The **lifetime ca-certificate** command takes effect when the local CA server certificate is first generated (that is, when you initially configure the local CA server and issue the **no shutdown** command). When the CA certificate expires, the configured lifetime value is used to generate the new CA certificate. You cannot change the lifetime value for existing CA certificates.
- You should configure the ASA to use an identity certificate to protect ASDM traffic and HTTPS traffic to the management interface. Identity certificates that are automatically generated with SCEP are regenerated after each reboot, so make sure that you manually install your own identity certificates. For an example of this procedure that applies only to SSL, see the following URL: http://www.cisco.com/en/US/products/ps6120/products_configuration_example09186a00809fcf91.shtml.
- The ASA and the AnyConnect clients can only validate certificates in which the X520Serialnumber field (the serial number in the Subject Name) is in PrintableString format. If the serial number format uses encoding such as UTF8, the certificate authorization will fail.
- Use only valid characters and values for certificate parameters when you import them on the ASA.
- To use a wildcard (*) symbol, make sure that you use encoding on the CA server that allows this character in the string value. Although RFC 5280 recommends using either a UTF8String or PrintableString, you should use UTF8String because PrintableString does not recognize the wildcard as a valid character. The ASA rejects the imported certificate if an invalid character or value is found during the import. For example:

```
ERROR: Failed to parse or verify imported certificate ciscoasa(config)# Read 162*H÷ytes
 as CA certificate:0U0= \Ivr"phÕV°3é¼þ0 CRYPTO_PKI(make trustedCerts list)
CERT-C: E ../cert-c/source/certlist.c(302): Error #711h
CRYPTO_PKI: Failed to verify the ID certificate using the CA certificate in trustpoint
 mm.
CERT-C: E ../cert-c/source/p7contnt.c(169): Error #703h
crypto_certc_pkcs7_extract_certs_and_crls failed (1795):
crypto_certc_pkcs7_extract_certs_and_crls failed
CRYPTO_PKI: status = 1795: failed to verify or insert the cert into storage
```

Configure Digital Certificates

The following topics explain how to configure digital certificates.

Configure Key Pairs

To create or remove key pairs, perform the following steps.

Procedure

Step 1 Generate one default, general-purpose RSA key pair.

crypto key generate rsa modulus 2048

Example:

```
ciscoasa(config)# crypto key generate rsa modulus 2048
```

The default key modulus is 2048, but you should specify the modulus explicitly to ensure you get the size you require. The key is named Default-RSA-Key.

For RSA keys, the modulus can be one of the following (in bits): 512, 768, 1024, 2048, 4096.

If you also want an Elliptic Curve Signature Algorithm (ECDSA) key, you can generate the Default-ECDSA-Key. The default length is 384, but you can also use 256 or 521.

crypto key generate ecdsa elliptic-curve 384

Step 2 (Optional) Create additional keys with unique names.

crypto key generate rsa label *key-pair-label* **modulus** *size*

crypto key generate ecdsa label *key-pair-label* **elliptic-curve** *size*

Example:

```
ciscoasa(config)# crypto key generate rsa label exchange modulus 2048
```

The label is referenced by the trustpoint that uses the key pair.

Step 3 Verify key pairs that you have generated.

show crypto key mypubkey {**rsa** | **ecdsa**}

Example:

```
ciscoasa/contexta(config)# show crypto mypubkey key rsa
```

Step 4 Save the key pair that you have generated.

write memory

Example:

```
ciscoasa(config)# write memory
```

Step 5 If necessary, remove existing key pairs so that you can generate new ones.

crypto key zeroize {**rsa** | **ecdsa**}

Example:

```
ciscoasa(config)# crypto key zeroize rsa
```

Step 6 (Optional) Archive the local CA server certificate and key pair.

copy

Example:

```
ciscoasa# copy LOCAL-CA-SERVER_0001.pl2 tftp://10.1.1.22/user6/
```

This command copies the local CA server certificate and key pair and all files from the ASA using either FTP or TFTP.

Note Make sure that you back up all local CA files as often as possible.

Example

The following example shows how to remove key pairs:

```
ciscoasa(config)# crypto key zeroize rsa
WARNING: All RSA keys will be removed.
WARNING: All device certs issued using these keys will also be removed.

Do you really want to remove these keys? [yes/no] y
```

Configure Trustpoints

To configure a trustpoint, perform the following steps:

Procedure

Step 1 Create a trustpoint that corresponds to the CA from which the ASA needs to receive a certificate.

crypto ca trustpoint *trustpoint-name*

Example:

```
ciscoasa/contexta(config)# crypto ca trustpoint Main
```

You enter the crypto ca trustpoint configuration mode, which controls CA-specific trustpoint parameters that you may configure starting in Step 3.

Step 2 Choose one of the following options:

- Request automatic enrollment using SCEP with the specified trustpoint and configure the enrollment URL.

  enrollment protocol scep *url*

  Example:

  ```
  ciscoasa/contexta(config-ca-trustpoint)# enrollment protocol scep url
  ```

```
http://10.29.67.142:80/certsrv/mscep/mscep.dll
```

- Request automatic enrollment using CMP with the specified trustpoint and configure the enrollment URL.

  enrollment protocol cmp*url*

  Example

  ```
  ciscoasa/ contexta(config-ca-trustpoint)# enrollment protocol cmp url
  http://10.29.67.142:80/certsrv/mscep/mscep.dll
  ```

- Request manual enrollment with the specified trustpoint by pasting the certificate received from the CA into the terminal.

  enrollment terminal

  ```
  ciscoasa/contexta(config-ca-trustpoint)# enrollment terminal
  ```

- Request self signed certificate.

  enrollment self

Step 3 Specify the available CRL configuration options.

revocation-check crl none

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# revocation-check crl
ciscoasa/contexta(config-ca-trustpoint)# revocation-check none
```

Note To enable either required or optional CRL checking, make sure that you configure the trustpoint for CRL management after obtaining certificates.

Step 4 Enable or disable the basic constraints extension and CA flag.

[**no**] **ca-check**

The basic constraints extension identifies whether the subject of the certificate is a Certificate Authority (CA), in which case the certificate can be used to sign other certificates. The CA flag is part of this extension. The presence of these items in a certificate indicates that the certificate's public key can be used to validate certificate signatures.

The **ca-check** command is enabled by default, so you need to enter this command only if you want to disable basic constraints and the CA flag.

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# no ca-check
```

Step 5 During enrollment, ask the CA to include the specified e-mail address in the Subject Alternative Name extension of the certificate.

email *address*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# email example.com
```

Step 6 (Optional) Specify a retry period in minutes, and applies only to SCEP enrollment.

enrollment retry period

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment retry period 5
```

Step 7 (Optional) Specify a maximum number of permitted retries, and applies only to SCEP enrollment.

enrollment retry count

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment retry period 2
```

Step 8 During enrollment, ask the CA to include the specified fully qualified domain name in the Subject Alternative Name extension of the certificate.

fqdn *fqdn*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# fqdn example.com
```

Step 9 During enrollment, ask the CA to include the IP address of the ASA in the certificate.

ip-address *ip-address*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# ip-address 10.10.100.1
```

Step 10 Specify the key pair whose public key is to be certified.

keypair *name*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# keypair exchange
```

Step 11 Configure OCSP URL overrides and trustpoints to use for validating OCSP responder certificates.

match certificate map-name override ocsp

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# match certificate examplemap override ocsp
```

Step 12 Disable the nonce extension on an OCSP request. The nonce extension cryptographically binds requests with responses to avoid replay attacks.

ocsp disable-nonce

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# ocsp disable-nonce
```

Step 13 Configure an OCSP server for the ASA to use to check all certificates associated with a trustpoint rather than the server specified in the AIA extension of the client certificate.

ocsp url

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# ocsp url
```

Step 14 Specify a challenge phrase that is registered with the CA during enrollment. The CA usually uses this phrase to authenticate a subsequent revocation request.

password *string*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# password mypassword
```

Step 15 Set one or more methods for revocation checking: CRL, OCSP, and none.

revocation check

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# revocation check
```

Step 16 During enrollment, ask the CA to include the specified subject DN in the certificate. If a DN string includes a comma, enclose the value string within double quotes (for example, O=”Company, Inc.”).

subject-name *X.500 name*

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# myname X.500 examplename
```

Step 17 During enrollment, ask the CA to include the ASA serial number in the certificate.

serial-number

Example:

```
ciscoasa/contexta(config-ca-trustpoint)# serial number JMX1213L2A7
```

Step 18 Save the running configuration.

write memory

Example:

```
ciscoasa/contexta(config)# write memory
```

Configure CRLs for a Trustpoint

To use mandatory or optional CRL checking during certificate authentication, you must configure CRLs for each trustpoint. To configure CRLs for a trustpoint, perform the following steps:

Procedure

Step 1 Enter crypto ca trustpoint configuration mode for the trustpoint whose CRL configuration you want to modify.

crypto ca trustpoint *trustpoint-name*

Example:

```
ciscoasa (config)# crypto ca trustpoint Main
```

Note Make sure that you have enabled CRLs before entering this command. In addition, the CRL must be available for authentication to succeed.

Step 2 Enter crl configuration mode for the current trustpoint.

crl configure

Example:

```
ciscoasa(config-ca-trustpoint)# crl configure
```

Tip To set all CRL configuration parameters to default values, use the **default** command. At any time during CRL configuration, reenter this command to restart the procedure.

Step 3 Choose one of the following to configure retrieval policy:

- CRLs are retrieved only from the CRL distribution points that are specified in authenticated certificates.

  policy cdp

  ```
  ciscoasa(config-ca-crl)# policy cdp
  ```

  Note SCEP retrieval is not supported by distribution points specified in certificates.

- CRLs are retrieved only from URLs that you configure.

  policy static

  ```
  ciscoasa(config-ca-crl)# policy static
  ```

- CRLs are retrieved from CRL distribution points specified in authenticated certificates and from URLs that you configure.

  policy both

  ```
  ciscoasa(config-ca-crl)# policy both
  ```

Step 4 If you used the **static** or **both** keywords when you configured the CRL policy, you must configure URLs for CRL retrieval. You can enter up to five URLs, ranked 1 through 5. The *n* argument is the rank assigned to the URL.

url *n url*

Example:

```
ciscoasa (config-ca-crl)# url 2 http://www.example.com
```

To remove a URL, use the **no url** *n* command.

Step 5 Specify HTTP, LDAP, or SCEP as the CRL retrieval method.

protocol http | ldap | scep

Example:

```
ciscoasa(config-ca-crl)# protocol http
```

Step 6 Configure how long the ASA caches CRLs for the current trustpoint. The *refresh-time* argument is the number of minutes that the ASA waits before considering a CRL stale.

cache-time *refresh-time*

Example:

```
ciscoasa(config-ca-crl)# cache-time 420
```

Step 7 Choose one of the following:

- Require the NextUpdate field to be present in CRLs. This is the default setting.

  enforcenextupdate

  ```
  ciscoasa(config-ca-crl)# enforcenextupdate
  ```

- Allow the NextUpdate field to be absent in CRLs.

  no enforcenextupdate

  ```
  ciscoasa(config-ca-crl)# no enforcenextupdate
  ```

Step 8 Identify the LDAP server to the ASA if LDAP is specified as the retrieval protocol. You can specify the server by DNS hostname or by IP address. You can also provide a port number if the server listens for LDAP queries on a port other than the default of 389.

ldap-defaults *server*

Example:

```
ciscoasa (config-ca-crl)# ldap-defaults ldap1
```

Note If you use a hostname instead of an IP address to specify the LDAP server, make sure that you have configured the ASA to use DNS.

Step 9 Allow CRL retrieval if the LDAP server requires credentials.

ldap-dn *admin-DN password*

Example:

```
ciscoasa (config-ca-crl)# ldap-dn cn=admin,ou=devtest,o=engineering c00lRunZ
```

Step 10 Retrieve the current CRL from the CA represented by the specified trustpoint and test the CRL configuration for the current trustpoint.

crypto ca crl request *trustpoint*

Example:

```
ciscoasa (config-ca-crl)# crypto ca crl request Main
```

Step 11 Save the running configuration.

write memory

Example:

```
ciscoasa (config)# write memory
```

Export or Import a Trustpoint Configuration

To export and import a trustpoint configuration, perform the following steps:

Procedure

Step 1 Export a trustpoint configuration with all associated keys and certificates in PKCS12 format.

crypto ca export *trustpoint*

Example:

```
ciscoasa(config)# crypto ca export Main
```

The ASA displays the PKCS12 data in the terminal. You can copy the data. The trustpoint data is password protected; however, if you save the trustpoint data in a file, make sure that the file is in a secure location.

Step 2 Import keypairs and issued certificates that are associated with a trustpoint configuration.

crypto ca import *trustpoint* **pkcs12**

Example:

```
ciscoasa(config)# crypto ca import Main pkcs12
```

The ASA prompts you to paste the text into the terminal in base 64 format. The key pair imported with the trustpoint is assigned a label that matches the name of the trustpoint that you create.

Note If an ASA has trustpoints that share the same CA, you can use only one of the trustpoints that share the CA to validate user certificates. To control which trustpoint that shares a CA is used for validation of user certificates issued by that CA, use the **support-user-cert-validation** keyword.

Examples

The following example exports PKCS12 data for the trustpoint Main with the passphrase Wh0zits:

```
ciscoasa(config)# crypto ca export Main pkcs12 Wh0zits

Exported pkcs12 follows:

[ PKCS12 data omitted ]

---End - This line not part of the pkcs12---
```

The following example manually imports PKCS12 data to the trustpoint Main with the passphrase Wh0zits:

```
ciscoasa (config)# crypto ca import Main pkcs12 Wh0zits

Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully
```

The following example manually imports a certificate for the trustpoint Main:

```
ciscoasa (config)# crypto ca import Main certificate
% The fully-qualified domain name in the certificate will be: securityappliance.example.com

Enter the base 64 encoded certificate.
End with a blank line or the word “quit” on a line by itself
[ certificate data omitted ]
quit
```

```
INFO: Certificate successfully imported
```

Configure CA Certificate Map Rules

You can configure rules based on the Issuer and Subject fields of a certificate. Using the rules you create, you can map IPsec peer certificates to tunnel groups with the **tunnel-group-map** command.

To configure a CA certificate map rule, perform the following steps:

Procedure

Step 1 Enter CA certificate map configuration mode for the rule you want to configure and specify the rule sequence number.

crypto ca certificate map [*map_name*]*sequence-number*

Example:

```
ciscoasa(config)# crypto ca certificate map test-map 10
```

If you do not specify the map name, the rule is added to the default map: DefaultCertificateMap. For each rule number, you can specify one or more fields to match.

Step 2 Specify the issuer name or subject name:

{**issuer-name** | **subject-name**} [**attr** *attribute*] *operator string*

Example:

```
ciscoasa(config-ca-cert-map)# issuer-name cn=asa.example.com
ciscoasa(config-ca-cert-map)# subject-name attr cn eq mycert
ciscoasa(config-ca-cert-map)# subject-name attr uid eq jcrichton
```

You can match the entire value, or specify the attributes that you want to match. The following are valid attributes:

- c—Country
- cn—Common Name
- dc—Domain Component
- dnq—DN Qualifier
- ea—Email Address
- genq—Generational Qualifier
- gn—Given Name
- i—Initials
- ip—IP Address
- l—Locality

- n—Name
- o—Organization Name
- ou—Organizational Unit
- ser—Serial Number
- sn—Surname
- sp—State/Province
- t—Title
- uid—User ID
- uname—Unstructured Name

The following are valid operators:

- eq—The field or attribute must be identical to the value given.
- ne—The field or attribute cannot be identical to the value given.
- co—Part or all of the field or attribute must match the value given.
- nc—No part of the field or attribute can match the value given.

Step 3 Specify the alternative subject name:

alt-subject-name *operator string*

Example:

```
ciscoasa(config-ca-cert-map)# alt-subject-name eq happydays
```

The following are valid operators:

- eq—The field must be identical to the value given.
- ne—The field cannot be identical to the value given.
- co—Part or all of the field must match the value given.
- nc—No part of the field can match the value given.

Step 4 Specify the extended key usage:

extended-key-usage *operator OID_string*

Example:

```
ciscoasa(config-ca-cert-map)# extended-key-usage nc clientauth
```

The following are valid operators:

- co—Part or all of the field must match the value given.
- nc—No part of the field can match the value given.

The following are valid OID strings:

- *string*—User-defined string.
- clientauth—Client Authentication (1.3.6.1.5.5.7.3.2)
- codesigning—Code Signing (1.3.6.1.5.5.7.3.3)
- emailprotection—Secure Email Protection (1.3.6.1.5.5.7.3.4)
- ocspsigning—OCSP Signing (1.3.6.1.5.5.7.3.9)
- serverauth—Server Authentication (1.3.6.1.5.5.7.3.1)
- timestamping—Time Stamping (1.3.6.1.5.5.7.3.8)

Obtain Certificates Manually

To obtain certificates manually, perform the following steps:

Before you begin

You must have already obtained a base-64 encoded CA certificate from the CA represented by the trustpoint.

Procedure

Step 1 Import the CA certificate for the configured trustpoint.

crypto ca authenticate *trustpoint*

Example:

```
ciscoasa(config)# crypto ca authenticate Main
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit

INFO: Certificate has the following attributes:
Fingerprint:     24b81433 409b3fd5 e5431699 8d490d34
Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.

% Certificate successfully imported
```

Whether a trustpoint requires that you manually obtain certificates is determined by the use of the **enrollment terminal** command when you configure the trustpoint.

Step 2 Enroll the ASA with the trustpoint.

crypto ca enroll *trustpoint*

Example:

```
ciscoasa(config)# crypto ca enroll Main
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be: securityappliance.example.com

% Include the device serial number in the subject name? [yes/no]: n

Display Certificate Request to terminal? [yes/no]: y
Certificate Request follows:

MIIBoDCCAQkCAQAwIzEhMB8GCSqGSIb3DQEJAhYSRmVyYWxQaXguY2lzY28uY29t
[ certificate request data omitted ]
jF4waw68eOxQxVmdgMWeQ+RbIOYmvt8g6hnBTrd0GdqjjVLt

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: n
```

This command generates a certificate for signing data and depending on the type of keys that you have configured, for encrypting data. If you use separate RSA keys for signing and encryption, the **crypto ca enroll** command displays two certificate requests, one for each key. If you use general-purpose RSA keys for both signing and encryption, the **crypto ca enroll** command displays one certificate request.

To complete enrollment, obtain a certificate for all certificate requests generated by the **crypto ca enroll** command from the CA represented by the applicable trustpoint. Make sure that the certificate is in base-64 format.

Step 3 Import each certificate you receive from the CA and make sure that you paste the certificate to the terminal in base-64 format.

crypto ca import *trustpoint* **certificate**

Example:

```
ciscoasa (config)# crypto ca import Main certificate
% The fully-qualified domain name in the certificate will be: securityappliance.example.com

Enter the base 64 encoded certificate.
End with a blank line or the word “quit” on a line by itself
[ certificate data omitted ]
quit
INFO: Certificate successfully imported
```

Step 4 Verify that the enrollment process was successful by displaying certificate details issued for the ASA and the CA certificate for the trustpoint.

show crypto ca certificate

Example:

```
ciscoasa(config)# show crypto ca  certificate Main
```

Step 5 Save the running configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

Step 6 Repeat these steps for each trustpoint that you configure for manual enrollment.

Obtain Certificates Automatically with SCEP

This section describes how to obtain certificates automatically using SCEP.

Before you begin

You must have already obtained a base-64 encoded CA certificate from the CA represented by the trustpoint.

Procedure

Step 1 Obtain the CA certificate for the configured trustpoint.

crypto ca authenticate *trustpoint*

Example:

```
ciscoasa/contexta(config)# crypto ca authenticate Main
```

When you configure the trustpoint, use of the **enrollment url** command determines whether or not you must obtain certificates automatically via SCEP.

Step 2 Enroll the ASA with the trustpoint. This command retrieves a certificate for signing data and depending on the type of keys that you have configured, for encrypting data. Before entering this command, contact the CA administrator, who may need to authenticate the enrollment request manually before the CA grants certificates.

crypto ca enroll *trustpoint*

Example:

```
ciscoasa/contexta(config)# crypto ca enroll Main
```

If the ASA does not receive a certificate from the CA within one minute (the default) of sending a certificate request, it resends the certificate request. The ASA continues sending a certificate request each minute until a certificate is received.

If the fully qualified domain name configured for the trustpoint is not identical to the fully qualified domain name of the ASA, including the case of the characters, a warning appears. To resolve this issue, exit the enrollment process, make any necessary corrections, and reenter the **crypto ca enroll** command.

Note If the ASA reboots after you have issued the **crypto ca enroll** command but before you have received the certificate, reenter the **crypto ca enroll** command and notify the CA administrator.

Step 3 Verify that the enrollment process was successful by displaying certificate details issued for the ASA and the CA certificate for the trustpoint.

show crypto ca certificate

Example:

```
ciscoasa/contexta(config)# show crypto ca certificate Main
```

Step 4 Save the running configuration.

write memory

Example:

```
ciscoasa/contexta(config)# write memory
```

Configure Proxy Support for SCEP Requests

To configure the ASA to authenticate remote access endpoints using third-party CAs, perform the following steps:

Procedure

Step 1 Enter tunnel-group ipsec-attributes configuration mode.

tunnel-group *name* **ipsec-attributes**

Example:

```
ciscoasa(config)# tunnel-group remotegrp ipsec-attributes
```

Step 2 Enable client services.

crypto ikev2 enable outside client-services port *portnumber*

Example:

```
ciscoasa(config-tunnel-ipsec)# crypto ikev2 enable outside client-services
```

The default port number is 443.

Note This command is needed only if you support IKEv2.

Step 3 Enter tunnel-group general-attributes configuration mode.

tunnel-group *name* **general-attributes**

Example:

```
ciscoasa(config)# tunnel-group 209.165.200.225 general-attributes
```

Step 4 Enable SCEP enrollment for the tunnel group.

scep-enrollment enable

Example:

```
ciscoasa(config-tunnel-general)# scep-enrollment enable
INFO: 'authentication aaa certificate' must be configured to complete setup of this option.
```

Step 5 Enter group-policy attributes configuration mode.

group-policy *name* **attributes**

Example:

```
ciscoasa(config)# group-policy FirstGroup attributes
```

Step 6 Enroll the SCEP CA for the group policy. Enter this command once per group policy to support a third-party digital certificate.

scep-forwarding-url value *URL*

Example:

```
ciscoasa(config-group-policy)# scep-forwarding-url value http://ca.example.com:80/
```

URL is the SCEP URL on the CA.

Step 7 Supply a common, secondary password when a certificate is unavailable for WebLaunch support of the SCEP proxy.

secondary-pre-fill-username clientless hide use-common-password *password*

Example:

```
ciscoasa(config)# tunnel-group remotegrp webvpn-attributes
ciscoasa(config-tunnel-webvpn)# secondary-pre-fill-username clientless hide
use-common-password secret
```

You must use the **hide** keyword to support the SCEP proxy.

For example, a certificate is not available to an endpoint requesting one. Once the endpoint has the certificate, AnyConnect disconnects, then reconnects to the ASA to qualify for a DAP policy that provides access to internal network resources.

Step 8 Hide the secondary prefill username for AnyConnect VPN sessions.

secondary-pre-fill-username ssl-client hide use-common-password *password*

Example:

```
ciscoasa(config-tunnel-webvpn)# secondary-pre-fill-username ssl-client hide
use-common-password secret
```

Despite the **ssl-client** keyword inherited from earlier releases, use this command to support AnyConnect sessions that use either IKEv2 or SSL.

You must use the **hide** keyword to support the SCEP proxy.

Step 9 Supply the username when a certificate is unavailable.

secondary-username-from-certificate {**use-entire-name** | **use-script** | {*primary_attr* [*secondary-attr*]}} [**no-certificate-fallback cisco-secure-desktop machine-unique-id**]

Example:

```
ciscoasa(config-tunnel-webvpn)# secondary-username-from-certificate CN no-certificate-fallback
 cisco-secure-desktop machine-unique-id
```

Configure the CA Certificate Lifetime

To configure the local CA server certificate lifetime, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Determine the expiration date to be included in the certificate. The default lifetime of a local CA certificate is three years.

lifetime ca-certificate *time*

Example:

```
ciscoasa(config-ca-server)# lifetime ca-certificate 365
```

Make sure that you limit the validity period of the certificate to less than the recommended end date of 03:14:08 UTC, January 19, 2038.

Step 3 (Optional) Reset the local CA certificate lifetime to the default value of three years.

no lifetime ca-certificate

Example:

```
ciscoasa(config-ca-server)# no lifetime ca-certificate
```

The local CA server automatically generates a replacement CA certificate 30 days before it expires, which allows the replacement certificate to be exported and imported onto any other devices for certificate validation of user certificates that have been issued by the local CA certificate after the current local CA certificate has expired. The following pre-expiration syslog message is generated:

```
%ASA-1-717049: Local CA Server certificate is due to expire in days days and a replacement
```

```
 certificate is available for export.
```

Note When notified of this automatic rollover, the administrator must make sure that the new local CA certificate is imported onto all required devices before it expires.

Configure the User Certificate Lifetime

To configure the user certificate lifetime, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Set the length of time that you want user certificates to remain valid.

lifetime certificate *time*

Example:

```
ciscoasa(config-ca-server)# lifetime certificate 60
```

Note Before a user certificate expires, the local CA server automatically initiates certificate renewal processing by granting enrollment privileges to the user several days ahead of the certificate expiration date, setting renewal reminders, and delivering an e-mail message that includes the enrollment username and OTP for certificate renewal. Make sure that you limit the validity period of the certificate to less than the recommended end date of 03:14:08 UTC, January 19, 2038.

Configure the CRL Lifetime

To configure the CRL lifetime, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Set the length of time that you want the CRL to remain valid.

lifetime crl *time*

Example:

```
ciscoasa(config-ca-server)# lifetime crl 10
```

The local CA updates and reissues the CRL each time that a user certificate is revoked or unrevoked, but if no revocation changes occur, the CRL is reissued automatically once each CRL lifetime. If you do not specify a CRL lifetime, the default time period is six hours.

Step 3 Force the issuance of a CRL at any time, which immediately updates and regenerates a current CRL to overwrite the existing CRL.

crypto ca server crl issue

Example:

```
ciscoasa(config-ca-server)# crypto ca server crl issue

A new CRL has been issued.
```

Note Do not use this command unless the CRL file has been removed in error or has been corrupted and must be regenerated.

Configure the Server Keysize

To configure the server keysize, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Specify the size of the public and private keys generated at user-certificate enrollment.

keysize server

Example:

```
ciscoasa(config-ca-server)# keysize server 2048
```

The key pair size options are 512, 768, 1024, 2048, 4096 bits, and the default value is 1024 bits.

Note After you have enabled the local CA, you cannot change the local CA keysize, because all issued certificates would be invalidated. To change the local CA keysize, you must delete the current local CA and reconfigure a new one.

Example

The following is sample output that shows two user certificates in the database.

```
Username: user1
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:45:52 UTC Fri Jan 4 2017
Certificates Issued:
serial:   0x71
issued:   12:45:52 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
Username: user2
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:27:59 UTC Fri Jan 4 2008
Certificates Issued:
serial:   0x2
issued:   12:27:59 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
<--- More --->
```

How to Set Up Specific Certificate Types

After you have established trusted certificates, you can begin other fundamental tasks such as establishing identity certificates or more advanced configurations such as establishing local CA or code signing certificates.

Before you begin

Read about digital certificate information and establish trusted certificates. CA certificates with no private key are used by all VPN protocols and webvpn, and are configured in trustpoints to validate incoming client certificates. Similarly, a trustpool is a list of trusted certificates used by webvpn features to validate proxied connections to https servers and to validate the smart-call-home certificate.

Procedure

A local CA allows VPN clients to enroll for certificates directly from the ASA. This advanced configuration converts the ASA into a CA. To configure CAs, refer to CA Certificates, on page 568.

What to do next

Set up a certificate expiration alert or monitor digital certificates and certificate management history.

CA Certificates

This page is where you manage CA certificates. The following topics explain what you can do.

Configure the Local CA Server

To configure the local CA server, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Specify the SMTP from-address, a valid e-mail address that the local CA uses as a from address when sending e-mail messages that deliver one-time passwords (OTPs) for an enrollment invitation to users.

smtp from-address *e-mail_address*

Example:

```
ciscoasa(config-ca-server) # smtp from-address SecurityAdmin@example.com
```

Step 3 (Optional) Specify the subject-name DN that is appended to each username on issued certificates.

subject-name-default *dn*

Example:

```
ciscoasa(config-ca-server)# subject-name-default cn=engineer, o=asc systems, c=”US”
```

The subject-name DN and the username combine to form the DN in all user certificates that are issued by the local CA server. If you do not specify a subject-name DN, you must specify the exact subject name DN to be included in a user certificate each time that you add a user to the user database.

Note Make sure that you review all optional parameters carefully before you enable the configured local CA, because you cannot change issuer-name and keysize server values after you enable the local CA for the first time.

Step 4 Create the self-signed certificate and associate it with the local CA on the ASA.

no shutdown

Example:

```
ciscoasa(config-ca-server)# no shutdown
```

The self-signed certificate key usage extension has key encryption, key signature, CRL signing, and certificate signing capabilities.

Note After the self-signed local CA certificate has been generated, to change any characteristics, you must delete the existing local CA server and completely recreate it.

The local CA server keeps track of user certificates, so the administrator can revoke or restore privileges as needed.

Example

The following example shows how to configure the local CA server using the predefined default values for all required parameters:

```
ciscoasa(config)# crypto ca server
ciscoasa(config-ca-server)# smtp from-address SecurityAdmin@example.com
ciscoasa(config-ca-server)# subject-name-default cn=engineer, o=asc Systems, c=US
ciscoasa(config-ca-server)# no shutdown
```

CA Server Management

Delete the Local CA Server

To delete an existing local CA server (either enabled or disabled), perform the following steps:

Procedure

Enter one of the following commands to remove an existing local CA server (either enabled or disabled):

- **no crypto ca server**

  Example

  ```
  ciscoasa(config)# no crypto ca server
  ```

- **clear configure crypto ca server**

  Example

  ```
  ciscoasa(config)# clear config crypto ca server
  ```

Note Deleting the local CA server removes the configuration from the ASA. After the configuration has been deleted, it is unrecoverable.

Make sure that you also delete the associated local CA server database and configuration files (that is, all files with the wildcard name, LOCAL-CA-SERVER.*).

Manage User Certificates

To change the certificate status, perform the following steps:

Procedure

Step 1 Select specific certificates by username or by certificate serial number in the **Manage User Certificates** pane.

Step 2 Choose one of the following options:

- Click **Revoke**to remove user access if the user certificate lifetime period runs out. The local CA also marks the certificate as revoked in the certificate database, automatically updates the information, and reissues the CRL.
- Select a revoked certificate and click **Unrevoke** to restore access. The local CA also marks the certificate as unrevoked in the certificate database, automatically updates the certificate information, and reissues an updated CRL.

Step 3 Click **Apply** when you are done to save your changes.

Enable the Local CA Server

To enable the local CA server, perform the following steps.

Before you begin

Before enabling the local CA server, you must first create a passphrase of at least seven characters to encode and archive a PKCS12 file that includes the local CA certificate and keypair to be generated. The passphrase unlocks the PKCS12 archive if the CA certificate or keypair is lost.

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Enable the local CA server.

no shutdown

Example:

```
ciscoasa(config-ca-server)# no shutdown
```

This command generates the local CA server certificate, keypair and necessary database files, and archives the local CA server certificate and keypair in a PKCS12 file. You must enter an 8-65 alphanumeric password. After initial startup, you can disable the local CA without being prompted for the password.

Step 3 Save the configuration to make sure that the local CA certificate and keypair are not lost after a reboot occurs.

write memory

Example:

```
ciscoasa(config)# write memory
```

Examples

The following example enables the local CA server:

```
ciscoasa(config)# crypto ca server
ciscoasa(config-ca-server)# no shutdown

% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit

Password: caserver

Re-enter password: caserver

Keypair generation process begin. Please wait...
```

The following is sample output that shows local CA server configuration and status:

```
Certificate Server LOCAL-CA-SERVER:
    Status: enabled
    State: enabled
    Server's configuration is locked (enter "shutdown" to unlock it)
    Issuer name: CN=wz5520-1-16
    CA certificate fingerprint/thumbprint: (MD5)
        76dd1439 ac94fdbc 74a0a89f cb815acc
    CA certificate fingerprint/thumbprint: (SHA1)
        58754ffd 9f19f9fd b13b4b02 15b3e4be b70b5a83
    Last certificate issued serial number: 0x6
    CA certificate expiration timer: 14:25:11 UTC Jan 16 2008
    CRL NextUpdate timer: 16:09:55 UTC Jan 24 2007
    Current primary storage dir: flash:
```

Customize the Local CA Server

To configure a customized local CA server, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Specify parameters that do not have default values.

issuer-name *DN-string*

Example:

```
ciscoasa(config-ca-server)# issuer-name cn=xx5520,cn=30.132.0.25,ou=DevTest,ou=QA,o=ASC
Systems
```

Step 3 Specify the e-mail address that is to be used as the From: field of all e-mail messages that are generated by the local CA server.

smtp from-address *e-mail_address*

Example:

```
ciscoasa(config-ca-server)# smtp from-address SecurityAdmin@example.com
```

Step 4 Customize the text that appears in the subject field of all e-mail messages sent from the local CA server.

smtp subject *subject-line*

Example:

```
ciscoasa(config-ca-server)# smtp subject Priority E-Mail: Enclosed Confidential Information
 is Required for Enrollment
```

Step 5 Specify an optional subject-name DN to be appended to a username on issued certificates.

subject-name-default *dn*

Example:

```
ciscoasa(config-ca-server)# subject-name default cn=engineer, o=ASC Systems, c=US
```

The default subject-name DN becomes part of the username in all user certificates issued by the local CA server.

The allowed DN attribute keywords are as follows:

- C = Country
- CN = Common Name
- EA = E-mail Address

- L = Locality
- O = Organization Name
- OU = Organization Unit
- ST = State/Province
- SN = Surname
- ST = State/Province

Note If you do not specify a subject-name-default to serve as a standard subject-name default, you must specify a DN each time that you add a user.

Disable the Local CA Server

To disable the local CA server, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Disable the local CA server.

shutdown

Example:

```
ciscoasa(config-ca-server)# shutdown
INFO: Local CA Server has been shutdown.
```

This command disables website enrollment, allows you to modify the local CA server configuration, and stores the current configuration and associated files. After initial startup, you can reenable the local CA without being prompted for the password.

Set Up External Local CA File Storage

To configure external local CA file storage, perform the following steps:

Procedure

Step 1 Access configuration mode for the specific file system type.

mount *name* **type**

Example:

```
ciscoasa(config)# mount mydata type cifs
```

Step 2 Mount a CIFS file system.

mount *name* **type cifs**

Example:

```
ciscoasa(config-mount-cifs)# mount mydata type cifs
server 10.1.1.10 share myshare
domain example.com
username user6
password ********
status enable
```

Note Only the user who mounts a file system can unmount it with the **no mount** command.

Step 3 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 4 Specify the location of *mydata*, the premounted CIFS file system to be used for the local CA server database.

database path *mount-name directory-path*

Example:

```
ciscoasa(config-ca-server)# database path mydata:newuser
```

This command establishes a path to the server and then specifies the local CA file or folder name to use for storage and retrieval. To return local CA file storage to the ASA flash memory, use the **no database path** command.

Note To secure stored local CA files on an external server requires a premounted file system of file type CIFS or FTP that is username-protected and password-protected.

Step 5 Save the running configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

For external local CA file storage, each time that you save the ASA configuration, user information is saved from the ASA to the premounted file system and file location, *mydata:newuser*.

For flash memory storage, user information is saved automatically to the default location for the start-up configuration.

Example

The following example shows the list of local CA files that appear in flash memory or in external storage:

```
ciscoasa(config-ca-server)# dir LOCAL* //
Directory of disk0:/LOCAL*

75     -rwx  32          13:07:49 Jan 20 2007  LOCAL-CA-SERVER.ser
77     -rwx  229         13:07:49 Jan 20 2007  LOCAL-CA-SERVER.cdb
69     -rwx  0           01:09:28 Jan 20 2007  LOCAL-CA-SERVER.udb
81     -rwx  232         19:09:10 Jan 20 2007  LOCAL-CA-SERVER.crl
72     -rwx  1603        01:09:28 Jan 20 2007  LOCAL-CA-SERVER.p12

127119360 bytes total (79693824 bytes free)
```

Download and Store CRLs

To download and store CRLs, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Open a port on an interface to make the CRL accessible from that interface. The specified interface and port are used to listen for incoming requests for the CRL.

publish-crl interface *interface* **port** *portnumber*

Example:

```
ciscoasa(config-ca-server)# publish-crl outside 70
```

The interface and optional port selections are as follows:

- inside—Name of interface/GigabitEthernet0/1

- management—Name of interface/ Management0/0
- outside—Name of interface/GigabitEthernet0/0
- Port numbers can range from 1-65535. TCP port 80 is the HTTP default port number.

Note If you do not specify this command, the CRL is not accessible from the CDP location, because this command is required to open an interface to download the CRL file.

The CDP URL can be configured to use the IP address of an interface, and the path of the CDP URL and the filename can also be configured (for example, http://10.10.10.100/user8/my_crl_file).

In this case, only the interface with that IP address configured listens for CRL requests, and when a request comes in, the ASA matches the path, /user8/my_crl_file to the configured CDP URL. When the path matches, the ASA returns the stored CRL file.

Note The protocol must be HTTP, so the prefix displayed is http://.

Step 3 Specify the CDP to be included in all issued certificates. If you do not configure a specific location for the CDP, the default URL location is http://*hostname.domain*/+CSCOCA+/asa_ca.crl.

cdp-url *url*

Example:

```
ciscoasa(config-ca-server)# cdp-url http://172.16.1.1/pathname/myca.crl
```

The local CA updates and reissues the CRL each time a user certificate is revoked or unrevoked. If no revocation changes occur, the CRL is reissued once each CRL lifetime.

If this command is set to serve the CRL directly from the local CA ASA, see Download and Store CRLs for instructions about opening a port on an interface to make the CRL accessible from that interface.

The CRL exists for other devices to validate the revocation of certificates issued by the local CA. In addition, the local CA tracks all issued certificates and status within its own certificate database. Revocation checking is performed when a validating party needs to validate a user certificate by retrieving the revocation status from an external server, which might be the CA that issued the certificate or a server designated by the CA.

Enrollment and User Management

Set Up Enrollment Parameters

To set up enrollment parameters, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Specify the number of hours that an issued OTP for the local CA enrollment page is valid. The default expiration time is 72 hours.

otp expiration *timeout*

Example:

```
ciscoasa(config-ca-server)# otp expiration 24
```

Note The user OTP to enroll for a certificate on the enrollment website is also used as the password to unlock the PKCS12 file that includes the issued certificate and keypair for the specified user.

Step 3 Specify the number of hours an already-enrolled user can retrieve a PKCS12 enrollment file.

enrollment-retrieval *timeout*

Example:

```
ciscoasa(config-ca-server)# enrollment-retrieval 120
```

This time period begins when the user is successfully enrolled. The default retrieval period is 24 hours. Valid values for the retrieval period range from 1 to 720 hours. The enrollment retrieval period is independent of the OTP expiration period.

After the enrollment retrieval time expires, the user certificate and keypair are no longer available. The only way a user may receive a certificate is for the administrator to reinitialize certificate enrollment and allow a user to log in again.

Add and Enroll Users

To add a user who is eligible for enrollment in the local CA database, perform the following steps:

Procedure

Step 1 Add a new user to the local CA database.

crypto ca server user-db add *username* [**dn** *dn*] [**email** *emailaddress*]

Example:

```
ciscoasa(config-ca-server)# crypto ca server user-db add user1 dn user1@example.com, Engineer,
 Example Company, US, email user1@example.com
```

The username argument is a string of 4-64 characters, which is the simple username for the user being added. The username can be an e-mail address, which then is used to contact the user as necessary for enrollment invitations.

The *dn* argument is the distinguished name, a global, authoritative name of an entry in the OSI Directory (X.500) (for example, cn=user1@example.com, cn=Engineer, o=Example Company, c=US).

The *e-mail-address* argument is the e-mail address of the new user to which OTPs and notices are to be sent.

Step 2 Provide user privileges to a newly added user.

crypto ca server user-db allow *user*

Example:

```
ciscoasa(config-ca-server)# crypto ca server user-db allow user
```

Step 3 Notify a user in the local CA database to enroll and download a user certificate, which automatically e-mails the OTP to that user.

crypto ca server user-db email-otp *username*

Example:

```
ciscoasa(config-ca-server)# crypto ca server user-db email-otp exampleuser1
```

Note When an administrator wants to notify a user through e-mail, the administrator must specify the e-mail address in the username field or in the e-mail field when adding that user.

Step 4 Show the issued OTP.

crypto ca server user-db show-otp

Example:

```
ciscoasa(config-ca-server)# crypto ca server user-db show-otp
```

Step 5 Set the enrollment time limit in hours. The default expiration time is 72 hours.

otp expiration *timeout*

Example:

```
ciscoasa(config-ca-server)# otp expiration 24
```

This command defines the amount of time that the OTP is valid for user enrollment. This time period begins when the user is allowed to enroll.

After a user enrolls successfully within the time limit and with the correct OTP, the local CA server creates a PKCS12 file, which includes a keypair for the user and a user certificate that is based on the public key from the keypair generated and the subject-name DN specified when the user is added. The PKCS12 file contents are protected by a passphrase, the OTP. The OTP can be handled manually, or the local CA can e-mail this file to the user to download after the administrator allows enrollment.

The PKCS12 file is saved to temporary storage with the name, *username.p12*. With the PKCS12 file in storage, the user can return within the enrollment-retrieval time period to download the PKCS12 file as many times as needed. When the time period expires, the PKCS12 file is removed from storage automatically and is no longer available to download.

Note If the enrollment period expires before the user retrieves the PKCS12 file that includes the user certificate, enrollment is not permitted.

Renew Users

To specify the timing of renewal notices, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Specifies the number of days (1-90) before the local CA certificate expires that an initial reminder to re-enroll is sent to certificate owners.

renewal-reminder *time*

Example:

```
ciscoasa(config-ca-server)# renewal-reminder 7
```

If a certificate expires, it becomes invalid. Renewal notices and the times they are e-mailed to users are variable, and can be configured by the administrator during local CA server configuration.

Three reminders are sent. An e-mail is automatically sent to the certificate owner for each of the three reminders, provided an e-mail address is specified in the user database. If no e-mail address exists for the user, a syslog message alerts you of the renewal requirement.

The ASA automatically grants certificate renewal privileges to any user who holds a valid certificate that is about to expire, as long as the user still exists in the user database. Therefore, if an administrator does not want to allow a user to renew automatically, the administrator must remove the user from the database before the renewal time period.

Restore Users

To restore a user and a previously revoked certificate that was issued by the local CA server, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Restore a user and unrevoke a previously revoked certificate that was issued by the local CA server.

crypto ca server unrevoke *cert-serial-no*

Example:

```
ciscoasa(config-ca-server)# crypto ca server unrevoke 782ea09f
```

The local CA maintains a current CRL with serial numbers of all revoked user certificates. This list is available to external devices and can be retrieved directly from the local CA if it is configured to do so with the **cdp-url** command and the **publish-crl** command. When you revoke (or unrevoke) any current certificate by certificate serial number, the CRL automatically reflects these changes.

Remove Users

To delete a user from the user database by username, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Remove a user from the user database and allow revocation of any valid certificates that were issued to that user.

crypto ca server user-db remove *username*

Example:

```
ciscoasa(config-ca-server)# crypto ca server user-db remove user1
```

Revoke Certificates

To revoke a user certificate, perform the following steps:

Procedure

Step 1 Enter local ca server configuration mode.

crypto ca server

Example:

```
ciscoasa(config)# crypto ca server
```

Step 2 Enter the certificate serial number in hexadecimal format.

crypto ca server revoke *cert-serial-no*

Example:

```
ciscoasa(config-ca-server)# crypto ca server revoke 782ea09f
```

This command marks the certificate as revoked in the certificate database on the local CA server and in the CRL, which is automatically reissued.

Note The password is also required if the certificate for the ASA needs to be revoked, so make sure that you record it and store it in a safe place.

Set a Certificate Expiration Alert (for Identity or CA Certificates)

ASA checks all the CA and ID certificates in the trust points for expiration once every 24 hours. If a certificate is nearing expiration, a syslog will be issued as an alert.

A CLI is provided to configure the reminder and recurrence intervals. By default, reminders start at 60 days prior to expiration and recur every 7 days. You can configure the interval at which reminders are sent and the number of days before the expiration at which the first alert is sent by using the following command:

```
[no] crypto ca alerts expiration [begin <days before expiration>] [repeat <days>]
```

Irrespective of the alerts configuration, a reminder is sent every day during the last week of expiration. The following **show** and **clear** commands have also been added:

```
clear conf crypto ca alerts
show run crypto ca alerts
```

In addition to the renewal reminder, if an already expired certificate is found in the configuration, a syslog is generated once every day to rectify the configuration by either renewing the certificate or removing the expired certificate.

For example, assume that the expiration alerts are configured to begin at 60 days and repeat every 6 days after that. If the ASA is rebooted at 40 days, an alert is sent on that day, and the next alert is sent on the 36th day.

Note Expiration checking is not done on trust pool certificates. The Local CA trust point is treated as a regular trustpoint for expiration checking too.

Monitoring Digital Certificates

See the following commands for monitoring digital certificate status:

- **show crypto ca server**

  This command shows local CA configuration and status.

- **show crypto ca server cert-db**

  This command shows user certificates issued by the local CA.

- **show crypto ca server certificate**

  This command shows local CA certificates on the console in base 64 format and the rollover certificate when available, including the rollover certificate thumb print for verification of the new certificate during import onto other devices.

- **show crypto ca server crl**

  This command shows CRLs.

- **show crypto ca server user-db**

  This command shows users and their status, which can be used with the following qualifiers to reduce the number of displayed records:

  - allowed. Shows only users currently allowed to enroll.
  - enrolled. Shows only users that are enrolled and hold a valid certificate
  - expired. Shows only users holding expired certificates.
  - on-hold. Lists only users without a certificate and not currently allowed to enroll.

- **show crypto ca server user-db allowed**

  This command shows users who are eligible to enroll.

- **show crypto ca server user-db enrolled**

  This command shows enrolled users with valid certificates.

- **show crypto ca server user-db expired**

  This command shows users with expired certificates.

- **show crypto ca server user-db on-hold**

  This command shows users without certificates who are not allowed to enroll.

- **show crypto key** *name of key*

  This command shows key pairs that you have generated.

- **show running-config**

  This command shows local CA certificate map rules.

Examples

The following example shows an RSA general-purpose key:

```
ciscoasa/contexta(config)# show crypto key mypubkey rsa
Key pair was generated at: 16:39:47 central Feb 10 2010
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 2048
 Storage: config
 Key Data:

  30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
  00ea2c38 df9c606e ddb7b08a e8b0a1a8 65592d85 0711cac5 fceddee1 fa494297
  525fffc0 90da8a4c e696e44e 0646c661 48b3602a 960d7a3a 52dae14a 5f983603
  e1f33e40 a6ce04f5 9a812894 b0fe0403 f8d7e05e aea79603 2dcd56cc 01261b3e
  93bff98f df422fb1 2066bfa4 2ff5d2a4 36b3b1db edaebf16 973b2bd7 248e4dd2
  071a978c 6e81f073 0c4cd57b db6d9f40 69dc2149 e755fb0f 590f2da8 b620efe6
  da6e8fa5 411a841f e72bb8ea cf4bdb79 f4e57ff3 a940ce3b 4a2c7052 56c1d17b
  af8fe2e2 e58718c6 ed1da0f0 1c6f36eb 79eb1aeb f098b5c4 79e07658 a52d8c7a
  51ceabfb f8ade096 7217cf2d 3728077e 89441d89 9bf5f875 c8d2db39 c858bb7a
  7d020301 0001
```

The following example shows the local CA CRL:

```
ciscoasa(config)# show crypto ca server crl
Certificate Revocation List:
    Issuer: cn=xx5520-1-3-2007-1
    This Update: 13:32:53 UTC Jan 4 2010
    Next Update: 13:32:53 UTC Feb 3 2010
    Number of CRL entries: 2
    CRL size: 270 bytes
Revoked Certificates:
    Serial Number: 0x6f
    Revocation Date: 12:30:01 UTC Jan 4 2010
    Serial Number: 0x47
    Revocation Date: 13:32:48 UTC Jan 4 2010
```

The following example shows one user on-hold:

```
ciscoasa(config)# show crypto ca server user-db on-hold
username: wilma101
email:    <None>
dn:       <None>
allowed:  <not allowed>
notified: 0
ciscoasa(config)#
```

The following example shows output of the **show running-config** command, in which local CA certificate map rules appear:

```
crypto ca certificate map 1
 issuer-name co asc
```

```
subject-name attr ou eq Engineering
```

History for Certificate Management

Table 23: History for Certificate Management

Feature Name	Platform Releases	Description
Certificate management	7.0(1)	Digital certificates (including CA certificates, identity certificates, and code signer certificates) provide digital identification for authentication. A digital certificate includes information that identifies a device or user, such as the name, serial number, company, department, or IP address. CAs are trusted authorities that “sign” certificates to verify their authenticity, thereby guaranteeing the identity of the device or user. CAs issue digital certificates in the context of a PKI, which uses public-key or private-key encryption to ensure security.
Certificate management	7.2(1)	We introduced the following commands: **issuer-name** *DN-string*, **revocation-check crl none**, **revocation-check crl**, **revocation-check none**. We deprecated the following commands: **crl** {**required** | **optional** | **nocheck**}.

Feature Name	Platform Releases	Description
Certificate management	8.0(2)	We introduced the following commands: **cdp-url**, **crypto ca server**, **crypto ca server crl issue**, **crypto ca server revoke** *cert-serial-no*, **crypto ca server unrevoke** *cert-serial-no*, **crypto ca server user-db add** *user* [**dn** *dn*] [**email** *e-mail-address*], **crypto ca server user-db allow** {*username* | **all-unenrolled** | **all-certholders**} [**display-otp**] [**email-otp**] [**replace-otp**], **crypto ca server user-db email-otp** {*username* | **all-unenrolled** | **all-certholders**}, **crypto ca server user-db remove** *username*, **crypto ca server user-db show-otp** {*username* | **all-certholders** | **all-unenrolled**}, **crypto ca server user-db write**, [**no**] **database path** *mount-name directory-path*, **debug crypto ca server** [*level*], **lifetime** {**ca-certificate** | **certificate** | **crl**} *time*, **no shutdown**, **otp expiration** *timeout*, **renewal-reminder** *time*, **show crypto ca server**, **show crypto ca server cert-db** [**user** *username* | **allowed** | **enrolled** | **expired** | **on-hold**] [**serial** *certificate-serial-number*], **show crypto ca server certificate**, **show crypto ca server crl**, **show crypto ca server user-db** [**expired** | **allowed** | **on-hold** | **enrolled**], **show crypto key** *name of key*, **show running-config**, **shutdown**.
SCEP proxy	8.4(1)	We introduced this feature, which provides secure deployment of device certificates from third-party CAs. We introduced the following commands: **crypto ikev2 enable outside client-services port** *portnumber*, **scep-enrollment enable**, **scep-forwarding-url value** *URL*, **secondary-pre-fill-username clientless hide use-common-password** *password*, **secondary-pre-fill-username ssl-client hide use-common-password** *password*, **secondary-username-from-certificate** {**use-entire-name** | **use-script** | {*primary_attr* [*secondary-attr*]}} [**no-certificate-fallback cisco-secure-desktop machine-unique-id**].

CHAPTER 21

ARP Inspection and the MAC Address Table for Transparent Firewall Mode

This chapter describes how to customize the MAC address table and configure ARP Inspection for bridge groups.

About ARP Inspection and the MAC Address Table

For interfaces in a bridge group, ARP inspection prevents a “man-in-the-middle” attack. You can also customize other ARP settings. You can customize the MAC address table for bridge groups, including adding a static ARP entry to guard against MAC spoofing.

ARP Inspection for Bridge Group Traffic

By default, all ARP packets are allowed between bridge group members. You can control the flow of ARP packets by enabling ARP inspection.

ARP inspection prevents malicious users from impersonating other hosts or routers (known as ARP spoofing). ARP spoofing can enable a “man-in-the-middle” attack. For example, a host sends an ARP request to the gateway router; the gateway router responds with the gateway router MAC address. The attacker, however, sends another ARP response to the host with the attacker MAC address instead of the router MAC address. The attacker can now intercept all the host traffic before forwarding it on to the router.

ARP inspection ensures that an attacker cannot send an ARP response with the attacker MAC address, so long as the correct MAC address and the associated IP address are in the static ARP table.

When you enable ARP inspection, the ASA compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table, and takes the following actions:

- If the IP address, MAC address, and source interface match an ARP entry, the packet is passed through.

- If there is a mismatch between the MAC address, the IP address, or the interface, then the ASA drops the packet.
- If the ARP packet does not match any entries in the static ARP table, then you can set the ASA to either forward the packet out all interfaces (flood), or to drop the packet.

Note The dedicated Management interface never floods packets even if this parameter is set to flood.

MAC Address Table

When you use bridge groups, the ASA learns and builds a MAC address table in a similar way as a normal bridge or switch: when a device sends a packet through the bridge group, the ASA adds the MAC address to its table. The table associates the MAC address with the source interface so that the ASA knows to send any packets addressed to the device out the correct interface. Because traffic between bridge group members is subject to the ASA security policy, if the destination MAC address of a packet is not in the table, the ASA does not flood the original packet on all interfaces as a normal bridge does. Instead, it generates the following packets for directly-connected devices or for remote devices:

- Packets for directly-connected devices—The ASA generates an ARP request for the destination IP address, so that it can learn which interface receives the ARP response.
- Packets for remote devices—The ASA generates a ping to the destination IP address so that it can learn which interface receives the ping reply.

The original packet is dropped.

Default Settings

- If you enable ARP inspection, the default setting is to flood non-matching packets.
- The default timeout value for dynamic MAC address table entries is 5 minutes.
- By default, each interface automatically learns the MAC addresses of entering traffic, and the ASA adds corresponding entries to the MAC address table.

Note ASA generates a reset packet to reset a connection that is denied by a stateful inspection engine. Here, the destination MAC address of the packet is not determined based on the ARP table lookup but instead it is taken directly from the packets (connections) that are being denied.

Guidelines for ARP Inspection and the MAC Address Table

- ARP inspection is only supported for bridge groups.

- MAC address table configuration is only supported for bridge groups.
- Bridge groups are only supported in transparent firewall mode.

Configure ARP Inspection and Other ARP Parameters

For transparent firewall mode bridge groups, you can enable ARP inspection. You can also configure other ARP parameters for both bridge groups and for routed mode interfaces.

Procedure

Step 1 Add static ARP entries according to Add a Static ARP Entry and Customize Other ARP Parameters, on page 589. ARP inspection compares ARP packets with static ARP entries in the ARP table, so static ARP entries are required for this feature. You can also configure other ARP parameters.

Step 2 (Transparent Mode Only) Enable ARP inspection according to Enable ARP Inspection, on page 590.

Add a Static ARP Entry and Customize Other ARP Parameters

By default for bridge groups, all ARP packets are allowed between bridge group member interfaces. You can control the flow of ARP packets by enabling ARP inspection. ARP inspection compares ARP packets with *static* ARP entries in the ARP table.

For routed interfaces, you can enter static ARP entries, but normally dynamic entries are sufficient. For routed interfaces, the ARP table is used to deliver packets to directly-connected hosts. Although senders identify a packet destination by an IP address, the actual delivery of the packet on Ethernet relies on the Ethernet MAC address. When a router or host wants to deliver a packet on a directly connected network, it sends an ARP request asking for the MAC address associated with the IP address, and then delivers the packet to the MAC address according to the ARP response. The host or router keeps an ARP table so it does not have to send ARP requests for every packet it needs to deliver. The ARP table is dynamically updated whenever ARP responses are sent on the network, and if an entry is not used for a period of time, it times out. If an entry is incorrect (for example, the MAC address changes for a given IP address), the entry needs to time out before it can be updated with the new information.

For transparent mode, the ASA only uses dynamic ARP entries in the ARP table for traffic to and from the ASA, such as management traffic.

You can also set the ARP timeout and other ARP behavior.

Procedure

Step 1 Add a static ARP entry:

arp *interface_name ip_address mac_address* [**alias**]

Example:

```
ciscoasa(config)# arp outside 10.1.1.1 0009.7cbe.2100
```

This example allows ARP responses from the router at 10.1.1.1 with the MAC address 0009.7cbe.2100 on the outside interface.

Specify **alias** in routed mode to enable proxy ARP for this mapping. If the ASA receives an ARP request for the specified IP address, then it responds with the ASA MAC address. This keyword is useful if you have devices that do not perform ARP, for example. In transparent firewall mode, this keyword is ignored; the ASA does not perform proxy ARP.

Step 2 Set the ARP timeout for dynamic ARP entries:

arp timeout *seconds*

Example:

```
ciscoasa(config)# arp timeout 5000
```

This field sets the amount of time before the ASA rebuilds the ARP table, between 60 to 4294967 seconds. The default is 14400 seconds. Rebuilding the ARP table automatically updates new host information and removes old host information. You might want to reduce the timeout because the host information changes frequently.

Step 3 Allow non-connected subnets:

arp permit-nonconnected

The ASA ARP cache only contains entries from directly-connected subnets by default. You can enable the ARP cache to also include non-directly-connected subnets. We do not recommend enabling this feature unless you know the security risks. This feature could facilitate denial of service (DoS) attack against the ASA; a user on any interface could send out many ARP replies and overload the ASA ARP table with false entries.

You may want to use this feature if you use:

- Secondary subnets.
- Proxy ARP on adjacent routes for traffic forwarding.

Enable ARP Inspection

This section describes how to enable ARP inspection for bridge groups.

Procedure

Enable ARP inspection:

arp-inspection *interface_name* **enable** [**flood** | **no-flood**]

Example:

```
ciscoasa(config)# arp-inspection outside enable no-flood
```

The **flood** keyword forwards non-matching ARP packets out all interfaces, and **no-flood** drops non-matching packets.

The default setting is to flood non-matching packets. To restrict ARP through the ASA to only static entries, then set this command to **no-flood**.

Customize the MAC Address Table for Transparent Mode Bridge Groups

This section describes how you can customize the MAC address table for bridge groups.

Add a Static MAC Address for Bridge Groups

Normally, MAC addresses are added to the MAC address table dynamically as traffic from a particular MAC address enters an interface. You can add static MAC addresses to the MAC address table. One benefit to adding static entries is to guard against MAC spoofing. If a client with the same MAC address as a static entry attempts to send traffic to an interface that does not match the static entry, then the ASA drops the traffic and generates a system message. When you add a static ARP entry (see Add a Static ARP Entry and Customize Other ARP Parameters, on page 589), a static MAC address entry is automatically added to the MAC address table.

To add a static MAC address to the MAC address table, perform the following steps.

Procedure

Add a static MAC address entry:

mac-address-table static *interface_name mac_address*

Example:

```
ciscoasa(config)# mac-address-table static inside 0009.7cbe.2100
```

The *interface_name* is the source interface.

Set the MAC Address Timeout

The default timeout value for dynamic MAC address table entries is 5 minutes, but you can change the timeout. To change the timeout, perform the following steps.

Procedure

Set the MAC address entry timeout:

mac-address-table aging-time *timeout_value*

Example:

```
ciscoasa(config)# mac-address-table aging-time 10
```

The *timeout_value* (in minutes) is between 5 and 720 (12 hours). 5 minutes is the default.

Configure MAC Address Learning

By default, each interface automatically learns the MAC addresses of entering traffic, and the ASA adds corresponding entries to the MAC address table. You can disable MAC address learning if desired, however, unless you statically add MAC addresses to the table, no traffic can pass through the ASA.

To configure MAC address learning, perform the following steps:

Procedure

Disable MAC address learning:

mac-learn *interface_name* **disable**

Example:

```
ciscoasa(config)# mac-learn inside disable
```

The **no** form of this command reenables MAC address learning.

The **clear configure mac-learn** command reenables MAC address learning on all interfaces.

Monitoring ARP Inspection and the MAC Address Table

- **show arp-inspection**

  Monitors ARP Inspection. Shows the current settings for ARP inspection on all interfaces.

- **show mac-address-table [*interface_name*]**

  Monitors the MAC address table. You can view the entire MAC address table (including static and dynamic entries for both interfaces), or you can view the MAC address table for an interface.

  The following is sample output from the **show mac-address-table** command that shows the entire table:

```
ciscoasa# show mac-address-table
interface        mac address           type         Time Left
-----------------------------------------------------------------------
outside     0009.7cbe.2100       static    -
inside     0010.7cbe.6101       static    -
inside     0009.7cbe.5101       dynamic    10
```

The following is sample output from the **show mac-address-table** command that shows the table for the inside interface:

```
ciscoasa# show mac-address-table inside
interface        mac address       type         Time Left
-----------------------------------------------------------------------
inside     0010.7cbe.6101       static    -
inside     0009.7cbe.5101       dynamic    10
```

History for ARP Inspection and the MAC Address Table

Feature Name	Platform Releases	Feature Information
ARP inspection	7.0(1)	ARP inspection compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table. This feature is available for Transparent Firewall Mode. We introduced the following commands: **arp**, **arp-inspection**, and **show arp-inspection**.
MAC address table	7.0(1)	You might want to customize the MAC address table for transparent mode. We introduced the following commands: **mac-address-table static**, **mac-address-table aging-time**, **mac-learn disable**, and **show mac-address-table**.
ARP cache additions for non-connected subnets	8.4(5)/9.1(2)	The ASA ARP cache only contains entries from directly-connected subnets by default. You can now enable the ARP cache to also include non-directly-connected subnets. We do not recommend enabling this feature unless you know the security risks. This feature could facilitate denial of service (DoS) attack against the ASA; a user on any interface could send out many ARP replies and overload the ASA ARP table with false entries. You may want to use this feature if you use: • Secondary subnets. • Proxy ARP on adjacent routes for traffic forwarding. We introduced the following command: **arp permit-nonconnected**.

PART V

IP Routing

CHAPTER 22

Routing Overview

This chapter describes how routing behaves within the ASA.

Path Determination

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement, such as path bandwidth, that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which include route information. Route information varies depending on the routing algorithm used.

Routing algorithms fill routing tables with a variety of information. Destination or next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular router representing the next hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop.

Routing tables also can include other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used.

Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology. A link-state advertisement, another example of a message sent between routers, informs other routers of the state of the sender links. Link information also can be used to build a complete picture of network topology to enable routers to determine optimal routes to network destinations.

Note Asymmetric routing is only supported for Active/Active failover in multiple context mode.

Supported Route Types

There are several route types that a router can use. The ASA uses the following route types:

- Static Versus Dynamic
- Single-Path Versus Multipath
- Flat Versus Hierarchical
- Link-State Versus Distance Vector

Static Versus Dynamic

Static routing algorithms are actually table mappings established by the network administrator. These mappings do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, they generally are considered unsuitable for large, constantly changing networks. Most of the dominant routing algorithms are dynamic routing algorithms, which adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.

Dynamic routing algorithms can be supplemented with static routes where appropriate. A router of last resort (a default route for a router to which all unroutable packets are sent), for example, can be designated to act as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

Single-Path Versus Multipath

Some sophisticated routing protocols support multiple paths to the same destination. Unlike single-path algorithms, these multipath algorithms permit traffic multiplexing over multiple lines. The advantages of multipath algorithms are substantially better throughput and reliability, which is generally called load sharing.

Flat Versus Hierarchical

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a flat routing system, the routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from non-backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more non-backbone routers to the final destination.

Routing systems often designate logical groups of nodes, called domains, autonomous systems, or areas. In hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain. In very large networks, additional hierarchical levels may exist, with routers at the highest hierarchical level forming the routing backbone.

The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns well. Most network communication occurs within small company groups (domains). Because intradomain routers need to know only about other routers within their domain, their routing algorithms can be simplified, and, depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

Link-State Versus Distance Vector

Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. In link-state algorithms, each router builds a picture of the entire network in its routing tables. Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers. Distance vector algorithms know only about their neighbors. Typically, link-state algorithms are used in conjunction with OSPF routing protocols.

Supported Internet Protocols for Routing

The ASA supports several Internet protocols for routing. Each protocol is briefly described in this section.

- Enhanced Interior Gateway Routing Protocol (EIGRP)

  EIGRP is a Cisco proprietary protocol that provides compatibility and seamless interoperation with IGRP routers. An automatic-redistribution mechanism allows IGRP routes to be imported into Enhanced IGRP, and vice versa, so it is possible to add Enhanced IGRP gradually into an existing IGRP network.

- Open Shortest Path First (OSPF)

  OSPF is a routing protocol developed for Internet Protocol (IP) networks by the interior gateway protocol (IGP) working group of the Internet Engineering Task Force (IETF). OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area includes an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

- Routing Information Protocol (RIP)

  RIP is a distance-vector protocol that uses hop count as its metric. RIP is widely used for routing traffic in the global Internet and is an interior gateway protocol (IGP), which means that it performs routing within a single autonomous system.

- Border Gateway Protocol (BGP)

  BGP is an interautonomous system routing protocol. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISP). Customers connect to ISPs, and ISPs use BGP to exchange customer and ISP routes. When BGP is used between autonomous systems (AS), the protocol is referred to as External BGP (EBGP). If a service provider is using BGP to exchange routes within an AS, then the protocol is referred to as Interior BGP (IBGP).

Routing Table

This section decribes how the routing table work.

How the Routing Table Is Populated

The ASA routing table can be populated by statically defined routes, directly connected routes, and routes discovered by the dynamic routing protocols. Because the ASA can run multiple routing protocols in addition to having static and connected routes in the routing table, it is possible that the same route is discovered or entered in more than one manner. When two routes to the same destination are put into the routing table, the one that remains in the routing table is determined as follows:

- If the two routes have different network prefix lengths (network masks), then both routes are considered unique and are entered into the routing table. The packet forwarding logic then determines which of the two to use.

  For example, if the RIP and OSPF processes discovered the following routes:

  - RIP: 192.168.32.0/24
  - OSPF: 192.168.32.0/19

  Even though OSPF routes have the better administrative distance, both routes are installed in the routing table because each of these routes has a different prefix length (subnet mask). They are considered different destinations and the packet forwarding logic determines which route to use.

- If the ASA learns about multiple paths to the same destination from a single routing protocol, such as RIP, the route with the better metric (as determined by the routing protocol) is entered into the routing table.

  Metrics are values associated with specific routes, ranking them from most preferred to least preferred. The parameters used to determine the metrics differ for different routing protocols. The path with the lowest metric is selected as the optimal path and installed in the routing table. If there are multiple paths to the same destination with equal metrics, load balancing is done on these equal cost paths.

- If the ASA learns about a destination from more than one routing protocol, the administrative distances of the routes are compared, and the routes with lower administrative distance are entered into the routing table.

Administrative Distances for Routes

You can change the administrative distances for routes discovered by or redistributed into a routing protocol. If two routes from two different routing protocols have the same administrative distance, then the route with the lower *default* administrative distance is entered into the routing table. In the case of EIGRP and OSPF routes, if the EIGRP route and the OSPF route have the same administrative distance, then the EIGRP route is chosen by default.

Administrative distance is a route parameter that the ASA uses to select the best path when there are two or more different routes to the same destination from two different routing protocols. Because the routing protocols have metrics based on algorithms that are different from the other protocols, it is not always possible to determine the best path for two routes to the same destination that were generated by different routing protocols.

Each routing protocol is prioritized using an administrative distance value. The following table shows the default administrative distance values for the routing protocols supported by the ASA.

Table 24: Default Administrative Distance for Supported Routing Protocols

Route Source	Default Administrative Distance
Connected interface	0
Static route	1
EIGRP Summary Route	5
External BGP	20
Internal EIGRP	90
OSPF	110
RIP	120
EIGRP external route	170
Internal and local BGP	200
Unknown	255

The smaller the administrative distance value, the more preference is given to the protocol. For example, if the ASA receives a route to a certain network from both an OSPF routing process (default administrative distance - 110) and a RIP routing process (default administrative distance - 120), the ASA chooses the OSPF route because OSPF has a higher preference. In this case, the router adds the OSPF version of the route to the routing table.

In this example, if the source of the OSPF-derived route was lost (for example, due to a power shutdown), the ASA would then use the RIP-derived route until the OSPF-derived route reappears.

The administrative distance is a local setting. For example, if you change the administrative distance of routes obtained through OSPF, that change would only affect the routing table for the ASA on which the command was entered. The administrative distance is not advertised in routing updates.

Administrative distance does not affect the routing process. The routing processes only advertise the routes that have been discovered by the routing process or redistributed into the routing process. For example, the RIP routing process advertises RIP routes, even if routes discovered by the OSPF routing process are used in the routing table.

Backup Dynamic and Floating Static Routes

A backup route is registered when the initial attempt to install the route in the routing table fails because another route was installed instead. If the route that was installed in the routing table fails, the routing table maintenance process calls each routing protocol process that has registered a backup route and requests them to reinstall the route in the routing table. If there are multiple protocols with registered backup routes for the failed route, the preferred route is chosen based on administrative distance.

Because of this process, you can create floating static routes that are installed in the routing table when the route discovered by a dynamic routing protocol fails. A floating static route is simply a static route configured with a greater administrative distance than the dynamic routing protocols running on the ASA. When the corresponding route discovered by a dynamic routing process fails, the static route is installed in the routing table.

How Forwarding Decisions Are Made

Forwarding decisions are made as follows:

- If the destination does not match an entry in the routing table, the packet is forwarded through the interface specified for the default route. If a default route has not been configured, the packet is discarded.
- If the destination matches a single entry in the routing table, the packet is forwarded through the interface associated with that route.
- If the destination matches more than one entry in the routing table, then the packet is forwarded out of the interface associated with the route that has the longer network prefix length.

For example, a packet destined for 192.168.32.1 arrives on an interface with the following routes in the routing table:

- 192.168.32.0/24 gateway 10.1.1.2
- 192.168.32.0/19 gateway 10.1.1.3

In this case, a packet destined to 192.168.32.1 is directed toward 10.1.1.2, because 192.168.32.1 falls within the 192.168.32.0/24 network. It also falls within the other route in the routing table, but 192.168.32.0/24 has the longest prefix within the routing table (24 bits verses 19 bits). Longer prefixes are always preferred over shorter ones when forwarding a packet.

Note Existing connections continue to use their established interfaces even if a new similar connection would result in different behavior due to a change in routes.

Dynamic Routing and Failover

Dynamic routes are synchronized on the standby unit when the routing table changes on the active unit. This means that all additions, deletions, or changes on the active unit are immediately propagated to the standby unit. If the standby unit becomes active in an active/standby ready Failover pair, it will already have an identical routing table as that of the former active unit because routes are synchronized as a part of the Failover bulk synchronization and continuous replication processes.

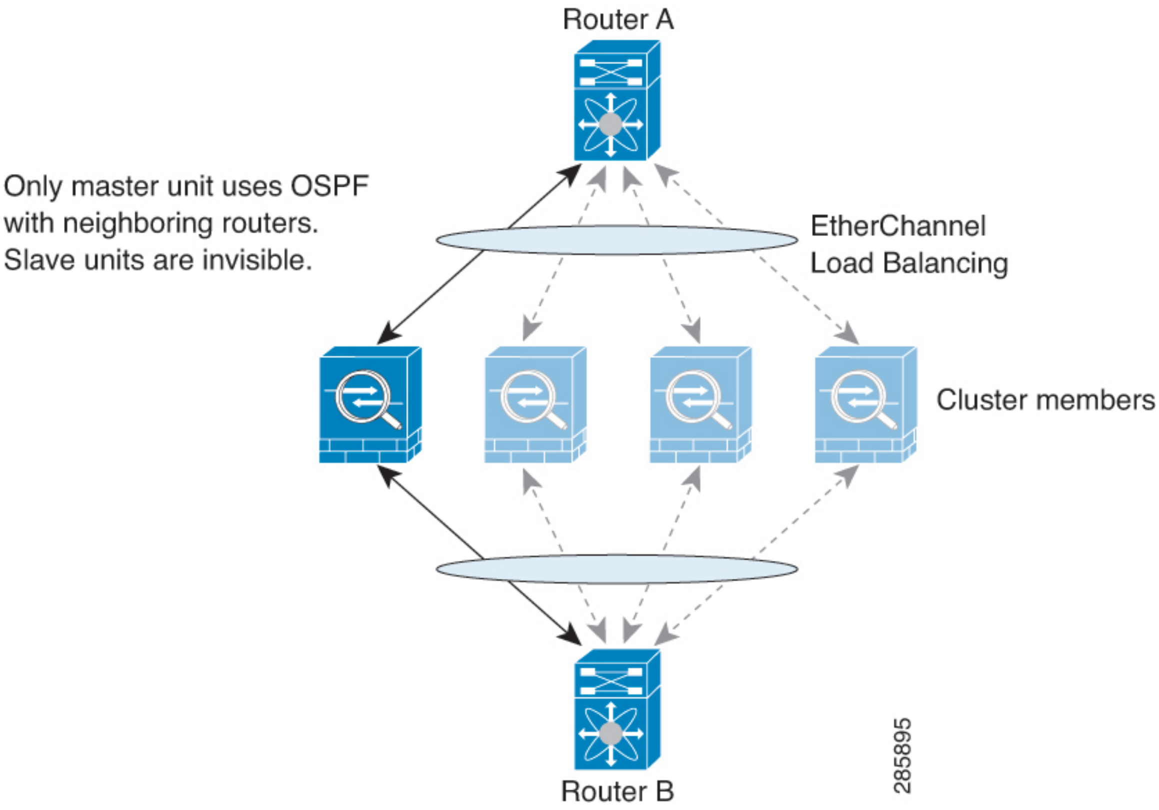
Dynamic Routing and Clustering

This section describes how to use dynamic routing with clustering.

Dynamic Routing in Spanned EtherChannel Mode

In Spanned EtherChannel mode: The routing process only runs on the control unit, and routes are learned through the control unit and replicated to data units. If a routing packet arrives at a data unit, it is redirected to the control unit.

Figure 52: Dynamic Routing in Spanned EtherChannel Mode


Router A
Only master unit uses OSPF
with neighboring routers.
Slave units are invisible.
EtherChannel
Load Balancing
Cluster members
Router B
285895

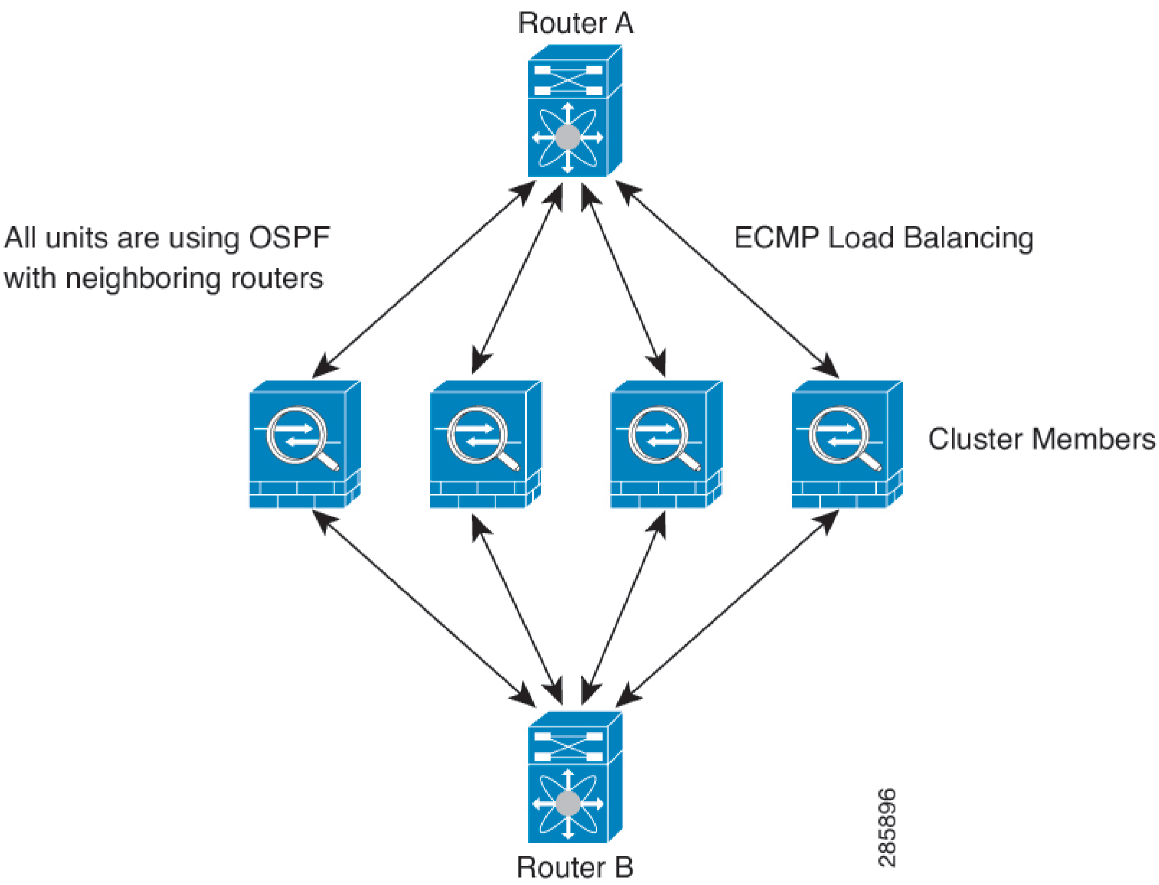
After the data unit learn the routes from the control unit, each unit makes forwarding decisions independently.

The OSPF LSA database is not synchronized from the control unit to data units. If there is a control unit switchover, the neighboring router will detect a restart; the switchover is not transparent. The OSPF process picks an IP address as its router ID. Although not required, you can assign a static router ID to ensure a consistent router ID is used across the cluster. See the OSPF Non-Stop Forwarding feature to address the interruption.

Dynamic Routing in Individual Interface Mode

In Individual interface mode, each unit runs the routing protocol as a standalone router, and routes are learned by each unit independently.

Figure 53: Dynamic Routing in Individual Interface Mode



In the above diagram, Router A learns that there are 4 equal-cost paths to Router B, each through an ASA. ECMP is used to load balance traffic between the 4 paths. Each ASA picks a different router ID when talking to external routers.

You must configure a cluster pool for the router ID so that each unit has a separate router ID.

EIGRP does not form neighbor relationships with cluster peers in individual interface mode.

Note If the cluster has multiple adjacencies to the same router for redundancy purposes, asymmetric routing can lead to unacceptable traffic loss. To avoid asymmetric routing, group all of these ASA interfaces into the same traffic zone. See Configure a Traffic Zone, on page 494.

Dynamic Routing in Multiple Context Mode

In multiple context mode, each context maintains a separate routing table and routing protocol databases. This enables you to configure OSPFv2 and EIGRP independently in each context. You can configure EIGRP in some contexts and OSPFv2 in the same or different contexts. In mixed context mode, you can enable any of the dynamic routing protocols in contexts that are in routed mode. RIP and OSPFv3 are not supported in multiple context mode.

The following table lists the attributes for EIGRP, OSPFv2, route maps used for distributing routes into OSPFv2 and EIGRP processes, and prefix lists used in OSPFv2 to filter the routing updates entering or leaving an area when they are used in multiple context mode:

EIGRP	OSPFv2	Route Maps and Prefix Lists
One instance is supported per context.	Two instances are supported per context.	N/A
It is disabled in the system context.		N/A
Two contexts may use the same or different autonomous system numbers.	Two contexts may use the same or different area IDs.	N/A
Shared interfaces in two contexts may have multiple EIGRP instances running on them.	Shared interfaces in two contexts may have multiple OSPF instances running on them.	N/A
The interaction of EIGRP instances across shared interfaces is supported.	The interaction of OSPFv2 instances across shared interfaces is supported.	N/A
All CLIs that are available in single mode are also available in multiple context mode.		
Each CLI has an effect only in the context in which it is used.		

Route Resource Management

A resource class called *routes* specifies the maximum number of routing table entries that can exist in a context. This resolves the problem of one context affecting the available routing table entries in another context and also allows you greater control over the maximum route entries per context.

Because there is no definitive system limit, you can only specify an absolute value for this resource limit; you may not use a percentage limit. Also, there are no minimum and maximum limits per context, so the default class does not change. If you add a new route for any of the static or dynamic routing protocols (connected, static, OSPF, EIGRP, and RIP) in a context and the resource limit for that context is exhausted, then the route addition fails and a syslog message is generated.

Equal-Cost Multi-Path (ECMP) Routing

The ASA supports Equal-Cost Multi-Path (ECMP) routing.

You can have up to 8 equal cost static or dynamic routes per interface. For example, you can configure multiple default routes on the outside interface that specify different gateways.

```
route outside 0 0 10.1.1.2
route outside 0 0 10.1.1.3
route outside 0 0 10.1.1.4
```

In this case, traffic is load-balanced on the outside interface between 10.1.1.2, 10.1.1.3, and 10.1.1.4. Traffic is distributed among the specified gateways based on an algorithm that hashes the source and destination IP addresses, incoming interface, protocol, source and destination ports.

ECMP Across Multiple Interfaces Using Traffic Zones

If you configure traffic zones to contain a group of interfaces, you can have up to 8 equal cost static or dynamic routes across up to 8 interfaces within each zone. For example, you can configure multiple default routes across three interfaces in the zone:

```
route outside1 0 0 10.1.1.2
route outside2 0 0 10.2.1.2
route outside3 0 0 10.3.1.2
```

Similarly, your dynamic routing protocol can automatically configure equal cost routes. The ASA load-balances traffic across the interfaces with a more robust load balancing mechanism.

When a route is lost, the device seamlessly moves the flow to a different route.

Disable Proxy ARP Requests

When a host sends IP traffic to another device on the same Ethernet network, the host needs to know the MAC address of the device. ARP is a Layer 2 protocol that resolves an IP address to a MAC address. A host sends an ARP request asking “Who is this IP address?” The device owning the IP address replies, “I own that IP address; here is my MAC address.”

Proxy ARP is used when a device responds to an ARP request with its own MAC address, even though the device does not own the IP address. The ASA uses proxy ARP when you configure NAT and specify a mapped address that is on the same network as the ASA interface. The only way traffic can reach the hosts is if the ASA uses proxy ARP to claim that the MAC address is assigned to destination mapped addresses.

Under rare circumstances, you might want to disable proxy ARP for NAT addresses.

If you have a VPN client address pool that overlaps with an existing network, the ASA by default sends proxy ARP requests on all interfaces. If you have another interface that is on the same Layer 2 domain, it will see the ARP requests and will answer with the MAC address of its interface. The result of this is that the return traffic of the VPN clients towards the internal hosts will go to the wrong interface and will get dropped. In this case, you need to disable proxy ARP requests for the interface on which you do not want them.

Procedure

Disable proxy ARP requests:

sysopt noproxyarp *interface*

Example:

```
ciscoasa(config)# sysopt noproxyarp exampleinterface
```

Display the Routing Table

Use the **show route** command to view the entries in the routing table.

```
ciscoasa# show route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.86.194.1 to network 0.0.0.0

S    10.1.1.0 255.255.255.0 [3/0] via 10.86.194.1, outside
C    10.86.194.0 255.255.254.0 is directly connected, outside
S*   0.0.0.0 0.0.0.0 [1/0] via 10.86.194.1, outside
```

CHAPTER 23

Static and Default Routes

This chapter describes how to configure static and default routes on the Cisco ASA.

About Static and Default Routes

To route traffic to a non-connected host or network, you must define a route to the host or network, either using static or dynamic routing. Generally, you must configure at least one static route: a default route for all traffic that is not routed by other means to a default network gateway, typically the next hop router.

Default Route

The simplest option is to configure a default static route to send all traffic to an upstream router, relying on the router to route the traffic for you. A default route identifies the gateway IP address to which the ASA sends all IP packets for which it does not have a learned or static route. A default static route is simply a static route with 0.0.0.0/0 (IPv4) or ::/0 (IPv6) as the destination IP address.

You should always define a default route.

Static Routes

You might want to use static routes in the following cases:

- Your networks use an unsupported router discovery protocol.
- Your network is small and you can easily manage static routes.
- You do not want the traffic or CPU overhead associated with routing protocols.
- In some cases, a default route is not enough. The default gateway might not be able to reach the destination network, so you must also configure more specific static routes. For example, if the default gateway is

outside, then the default route cannot direct traffic to any inside networks that are not directly connected to the ASA.

- You are using a feature that does not support dynamic routing protocols.

Route to null0 Interface to Drop Unwanted Traffic

Access rules let you filter packets based on the information contained in their headers. A static route to the null0 interface is a complementary solution to access rules. You can use a null0 route to forward unwanted or undesirable traffic so the traffic is dropped.

Static null0 routes have a favorable performance profile. You can also use static null0 routes to prevent routing loops. BGP can leverage the static null0 route for Remotely Triggered Black Hole routing.

Route Priorities

- Routes that identify a specific destination take precedence over the default route.
- When multiple routes exist to the same destination (either static or dynamic), then the administrative distance for the route determines priority. Static routes are set to 1, so they typically are the highest priority routes.
- When you have multiple static routes to the same destination with the same administrative distance, see Equal-Cost Multi-Path (ECMP) Routing, on page 605.
- For traffic emerging from a tunnel with the Tunneled option, this route overrides any other configured or learned default routes.

Transparent Firewall Mode Routes

For traffic that originates on the ASA and is destined through a bridge group member interface for a non-directly connected network, you need to configure either a default route or static routes so the ASA knows out of which bridge group member interface to send traffic. Traffic that originates on the ASA might include communications to a syslog server or SNMP server. If you have servers that cannot all be reached through a single default route, then you must configure static routes. For transparent mode, you cannot specify the BVI as the gateway interface; only member interfaces can be used. See #unique_851 for more information.

Static Route Tracking

One of the problems with static routes is that there is no inherent mechanism for determining if the route is up or down. They remain in the routing table even if the next hop gateway becomes unavailable. Static routes are only removed from the routing table if the associated interface on the ASA goes down.

The static route tracking feature provides a method for tracking the availability of a static route and installing a backup route if the primary route should fail. For example, you can define a default route to an ISP gateway and a backup default route to a secondary ISP in case the primary ISP becomes unavailable.

The ASA implements static route tracking by associating a static route with a monitoring target host on the destination network that the ASA monitors using ICMP echo requests. If an echo reply is not received within a specified time period, the host is considered down, and the associated route is removed from the routing table. An untracked backup route with a higher metric is used in place of the removed route.

When selecting a monitoring target, you need to make sure that it can respond to ICMP echo requests. The target can be any network object that you choose, but you should consider using the following:

- The ISP gateway (for dual ISP support) address
- The next hop gateway address (if you are concerned about the availability of the gateway)
- A server on the target network, such as a syslog server, that the ASA needs to communicate with
- A persistent network object on the destination network

Note A PC that may be shut down at night is not a good choice.

You can configure static route tracking for statically defined routes or default routes obtained through DHCP or PPPoE. You can only enable PPPoE clients on multiple interfaces with route tracking configured.

Guidelines for Static and Default Routes

Firewall Mode and Bridge Groups

- In transparent mode, static routes must use the bridge group member interface as the gateway; you cannot specify the BVI.
- Static route tracking is not supported for bridge group member interfaces.

IPv6

- Static route tracking is not supported for IPv6.

Clustering and Multiple Context Mode

- In clustering, static route tracking is only supported on the primary unit.
- Static route tracking is not supported in multiple context mode.

Configure Default and Static Routes

At a minimum, you should configure a default route. You may need to configure static routes as well. In this section we will configure a default route, configure a static route and track a static route.

Configure a Default Route

A default route is simply a static route with 0.0.0.0/0 as the destination IP address. You should always have a default route, either configured manually with this procedure, or derived from a DHCP server or other routing protocol.

Before you begin

See the following guidelines for the Tunneled option:

- Do not enable unicast RPF (**ip verify reverse-path** command) on the egress interface of a tunneled route, because this setting causes the session to fail.
- Do not enable TCP intercept on the egress interface of the tunneled route, because this setting causes the session to fail.
- Do not use the VoIP inspection engines (CTIQBE, H.323, GTP, MGCP, RTSP, SIP, SKINNY), the DNS inspection engine, or the DCE RPC inspection engine with tunneled routes, because these inspection engines ignore the tunneled route.
- You cannot define more than one default route with the tunneled option.
- ECMP for tunneled traffic is not supported.
- Tunneled routes are not supported for bridge groups, which do not support VPN termination for through traffic.

Procedure

Add a default route:

IPv4:

route *if_name* **0.0.0.0 0.0.0.0** *gateway_ip* [**distance**] [**tunneled**]

IPv6:

ipv6 route *if_name* **::/0** *gateway_ip* [**distance**] [**tunneled**]

Example:

```
ciscoasa(config)# route outside 0.0.0.0 0.0.0.0 192.168.2.4
ciscoasa(config)# route inside 0.0.0.0 0.0.0.0 10.1.2.3 tunneled
ciscoasa(config)# ipv6 route inside ::/0 3FFE:1100:0:CC00::1
```

The *if_name* is the interface through which you want to send the specific traffic. For transparent mode, specify a bridge group member interface name.

The *distance* argument is the administrative distance for the route, between 1 and 254. The default is **1** if you do not specify a value. Administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connected routes. The default administrative distance for routes discovered by OSPF is 110. If a static route has the same administrative distance as a dynamic route, the static routes take precedence. Connected routes always take precedence over static or dynamically discovered routes.

Note For through-the-box traffic, if you have two default routes configured on different interfaces that have different metrics, the connection to the ASA that is made from the higher metric interface fails, but connections to the ASA from the lower metric interface succeed as expected.

You can define a separate default route for VPN traffic if you want your VPN traffic to use a different default route than your non VPN traffic using the **tunneled** keyword. For example, traffic incoming from VPN connections can be easily directed towards internal networks, while traffic from internal networks can be directed towards the outside. When you create a default route with the tunneled option, all traffic from a tunnel terminating on the ASA that cannot be routed using learned or static routes, is sent to this route. This option is not supported for bridge groups.

Tip You can enter **0 0** instead of **0.0.0.0 0.0.0.0** for the destination network address and mask, as shown in the following example: **route outside 0 0 192.168.2.4**

Configure a Static Route

A static route defines where to send traffic for specific destination networks.

Procedure

Add a static route:

IPv4:

route *if_name dest_ip mask gateway_ip* [**distance**]

IPv6:

ipv6 route *if_name dest_ipv6_prefix*/*prefix_length gateway_ip* [**distance**]

Example:

```
ciscoasa(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
ciscoasa(config)# ipv6 route outside 2001:DB8:1::0/32 2001:DB8:0:CC00::1
```

The *if_name* is the interface through which you want to send the specific traffic. To drop unwanted traffic, enter the **null0** interface. For transparent mode, specify a bridge group member interface name.

The *dest_ip* and *mask* or *dest_ipv6_prefix*/*prefix_length* arguments indicate the IP address for the destination network and the *gateway_ip* argument is the address of the next-hop router. The addresses you specify for the static route are the addresses that are in the packet before entering the ASA and performing NAT.

The *distance* argument is the administrative distance for the route. The default is **1** if you do not specify a value. Administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connected routes. The default administrative distance for routes discovered by OSPF is 110. If a static route has the same administrative distance as a dynamic route, the static route takes precedence. Connected routes always take precedence over static or dynamically discovered routes.

Example

The following example shows static routes for 3 networks that go to the same gateway, and another network that goes to a separate gateway:

```
route outside 10.10.10.0 255.255.255.0 192.168.1.1
route outside 10.10.20.0 255.255.255.0 192.168.1.1
route outside 10.10.30.0 255.255.255.0 192.168.1.1
route inside 10.10.40.0 255.255.255.0 10.1.1.1
```

Configure Static Route Tracking

To configure static route tracking, complete the following steps.

Procedure

Step 1 Define the monitoring process:

sla monitor sla_id

Example:

```
ciscoasa(config)# sla monitor 5
ciscoasa(config-sla-monitor)#
```

Step 2 Specify the monitoring protocol, the target host on the tracked network, and the interface through which you reach the network:

type echo protocol ipicmpecho target_ip **interface** if_name

Example:

```
ciscoasa(config-sla-monitor)# type echo protocol ipicmpecho 172.29.139.134
ciscoasa(config-sla-monitor-echo)#
```

The *target_ip* argument is the IP address of the network object whose availability the tracking process monitors. While this object is available, the tracking process route is installed in the routing table. When this object becomes unavailable, the tracking process removes the route and the backup route is used in its place.

Step 3 (Optional) Configure monitoring options. See the command reference for the following commands: **frequency, num-packets, request-data-size, threshold, timeout,** and **tos**.

Step 4 Schedule the monitoring process:

sla monitor schedule sla_id [**life** {**forever** | seconds}] [**start-time** {hh:mm [:ss] [month day | day month] | **pending** | **now** | **after** hh:mm:ss}] [**ageout** seconds] [**recurring**]

Example:

```
ciscoasa(config)# sla monitor schedule 5 life forever start-time now
```

Typically, you will use the **sla monitor schedule** *sla_id* **life forever start-time now** command for the monitoring schedule, and allow the monitoring configuration to determine how often the testing occurs.

However, you can schedule this monitoring process to begin in the future and to only occur at specified times.

Step 5 Associate a tracked static route with the SLA monitoring process:

track track_id **rtr** sla_id **reachability**

Example:

```
ciscoasa(config)# track 6 rtr 5 reachability
```

The *track_id* argument is a tracking number you assign with this command. The *sla_id* argument is the ID number of the SLA process.

Step 6 Track one of the following route types:

- Static route:

  route if_name dest_ip mask gateway_ip [distance] **track** track_id

  Example:

  ```
  ciscoasa(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1 track 6
  ```

  You cannot use the **tunneled** option.

- Default route obtained through DHCP:

  ```
  interface interface_id
    dhcp client route track track_id
    ip address dhcp setroute
  ```

- Default route obtained through PPPoE:

  ```
  interface interface_id
    pppoe client route track track_id
    ip address pppoe setroute
  ```

Step 7 Create an untracked backup route.

The backup route is a static route to the same destination as the tracked route, but through a different interface or gateway. You must assign this route a higher administrative distance (metric) than your tracked route.

Monitoring a Static or Default Route

- **show route**

  Displays the routing table.

Examples for Static or Default Routes

The following example shows how to create a static route that sends all traffic destined for 10.1.1.0/24 to the router 10.1.2.45, which is connected to the inside interface, defines three equal cost static routes that direct traffic to three different gateways on the dmz interface, and adds a default route for tunneled traffic and one for regular traffic.

```
route inside 10.1.1.0 255.255.255.0 10.1.2.45
route dmz 10.10.10.0 255.255.255.0 192.168.2.1
route dmz 10.10.10.0 255.255.255.0 192.168.2.2
route dmz 10.10.10.0 255.255.255.0 192.168.2.3
route outside 0 0 209.165.201.1
route inside 0 0 10.1.2.45 tunneled
```

History for Static and Default Routes

Table 25: Feature History for Static and Default Routes

Feature Name	Platform Releases	Feature Information
Static Route Tracking	7.2(1)	The static route tracking feature provides a method for tracking the availability of a static route and installing a backup route if the primary route should fail. We introduced the following commands: **clear configure sla, frequency, num-packets, request-data-size, show sla monitor, show running-config sla, sla monitor, sla monitor schedule, threshold, timeout, tos, track rtr**
Static null0 route to drop traffic	9.2(1)	Sending traffic to a null0 interface results in dropping the packets destined to the specified network. This feature is useful in configuring Remotely Triggered Black Hole (RTBH) for BGP. We modified the following command: **route**.

CHAPTER 24

Policy Based Routing

This chapter describes how to configure the Cisco ASA to support policy based routing (PBR). The following sections describe policy based routing, guidelines for PBR, and configuration for PBR.

About Policy Based Routing

Traditional routing is destination-based, meaning packets are routed based on destination IP address. However, it is difficult to change the routing of specific traffic in a destination-based routing system. With Policy Based Routing (PBR), you can define routing based on criteria other than destination network—PBR lets you route traffic based on source address, source port, destination address, destination port, protocol, or a combination of these.

Policy Based Routing:

- Lets you provide Quality of Service (QoS) to differentiated traffic.
- Lets you distribute interactive and batch traffic across low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths.
- Allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections.

Policy Based Routing can implement QoS by classifying and marking traffic at the network edge, and then using PBR throughout the network to route marked traffic along a specific path. This permits routing of packets originating from different sources to different networks, even when the destinations are the same, and it can be useful when interconnecting several private networks.

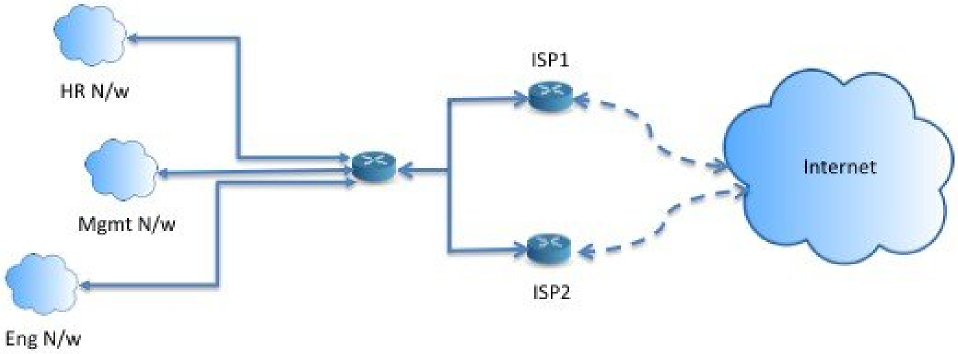
Why Use Policy Based Routing?

Consider a company that has two links between locations: one a high-bandwidth, low-delay expensive link, and the other a low-bandwidth, higher-delay, less-expensive link. While using traditional routing protocols, the higher-bandwidth link would get most, if not all, of the traffic sent across it based on the metric savings obtained by the bandwidth and/or delay (using EIGRP or OSPF) characteristics of the link. PBR allows you to route higher priority traffic over the high-bandwidth/low-delay link, while sending all other traffic over the low-bandwidth/high-delay link.

Some applications of policy based routing are:

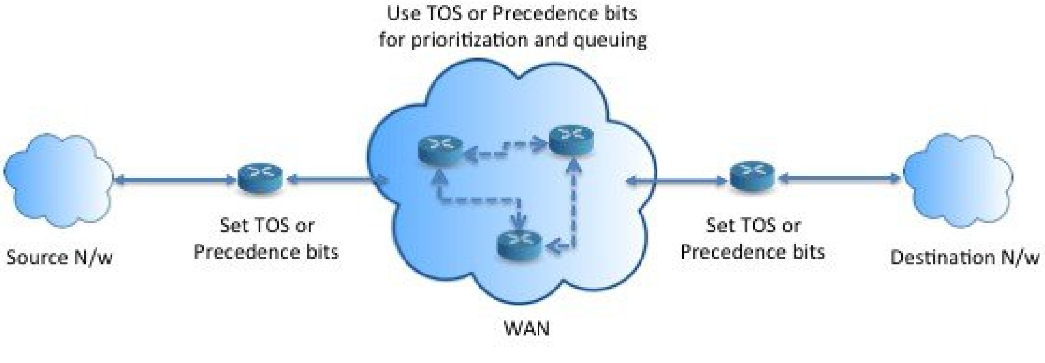
Equal-Access and Source-Sensitive Routing

In this topology, traffic from HR network & Mgmt network can be configured to go through ISP1 and traffic from Eng network can be configured to go through ISP2. Thus, policy based routing enables the network administrators to provide equal-access and source-sensitive routing, as shown here.



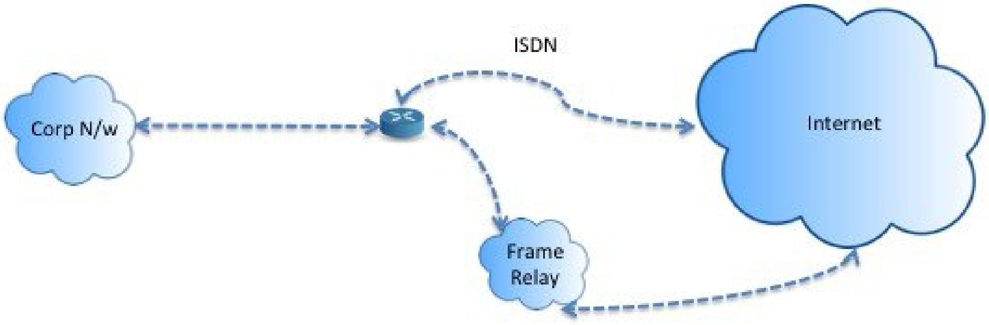
Quality of Service

By tagging packets with policy based routing, network administrators can classify the network traffic at the perimeter of the network for various classes of service and then implementing those classes of service in the core of the network using priority, custom or weighted fair queuing (as shown in the figure below). This setup improves network performance by eliminating the need to classify the traffic explicitly at each WAN interface in the core of backbone network.



Cost Saving

An organization can direct the bulk traffic associated with a specific activity to use a higher-bandwidth high-cost link for a short time and continues basic connectivity over a lower-bandwidth low-cost link for interactive traffic by defining the topology, as show here.



Load Sharing

In addition to the dynamic load-sharing capabilities offered by ECMP load balancing, network administrators can now implement policies to distribute traffic among multiple paths based on the traffic characteristics.

As an example, in the topology depicted in the Equal-Access Source Sensitive Routing scenario, an administrator can configure policy based routing to load share the traffic from HR network through ISP1 and traffic from Eng network through ISP2.

Implementation of PBR

The ASA uses ACLs to match traffic and then perform routing actions on the traffic. Specifically, you configure a route map that specifies an ACL for matching, and then you specify one or more actions for that traffic. Finally, you associate the route map with an interface where you want to apply PBR on all incoming traffic

Guidelines for Policy Based Routing

Firewall Mode

Supported only in routed firewall mode. Transparent firewall mode is not supported.

Per-flow Routing

Since the ASA performs routing on a per-flow basis, policy routing is applied on the first packet and the resulting routing decision is stored in the flow created for the packet. All subsequent packets belonging to the same connection simply match this flow and are routed appropriately.

PBR Policies Not Applied for Output Route Look-up

Policy Based Routing is an ingress-only feature; that is, it is applied only to the first packet of a new incoming connection, at which time the egress interface for the forward leg of the connection is selected. Note that PBR will not be triggered if the incoming packet belongs to an existing connection, or if NAT is applied.

Clustering

- Clustering is supported.
- In a cluster scenario, without static or dynamic routes, with ip-verify-reverse path enabled, asymmetric traffic may get dropped. So disabling ip-verify-reverse path is recommended.

Additional Guidelines

All existing route map related configuration restrictions and limitations will be carried forward.

Configure Policy Based Routing

A route map is comprised of one or more route-map statements. Each statement has a sequence number, as well as a permit or deny clause. Each route-map statement contains match and set commands. The match command denotes the match criteria to be applied on the packet. The set command denotes the action to be taken on the packet.

- When multiple next-hops or interfaces are configured as a set action, all options are evaluated one after the other until a valid usable option is found. No load balancing will be done among the configured multiple options.
- The verify-availability option is not supported in multiple context mode.

Procedure

Step 1 Define a standard or extended access-list:

access-list *name* **standard** {**permit** | **deny**} {**any4** | **host** *ip_address* | *ip_address mask*}

access-list *name* **extended** {**permit** | **deny**} *protocol source_and_destination_arguments*

Example:

```
ciscoasa(config)# access-list testacl extended permit ip
10.1.1.0 255.255.255.0 10.2.2.0 255.255.255.0
```

If you use a standard ACL, matching is done on the destination address only. If you use an extended ACL, you can match on source, destination, or both.

IPv6 ACLs are not supported.

Step 2 Create a route map entry:

route-map *name* {**permit** | **deny**} [*sequence_number*]

Example:

```
ciscoasa(config)# route-map testmap permit 12
```

Route map entries are read in order. You can identify the order using the *sequence_number* argument, or the ASA uses the order in which you add route map entries.

The ACL also includes its own permit and deny statements. For Permit/Permit matches between the route map and the ACL, the Policy Based Routing processing continues. For Permit/Deny matches, processing ends for this route map, and other route maps are checked. If the result is still Permit/Deny, then the regular routing table is used. For Deny/Deny matches, the Policy Based Routing processing continues.

Note When a route-map is configured without a permit or deny action and without a sequence-number, it by default will assume the action as permit and sequence-number as 10.

Step 3 Define the match criteria to be applied using an access-list:

match ip address *access-list_name* [*access-list_name...*]

Example:

```
ciscoasa(config-route-map)# match ip address testacl
```

Step 4 Configure one or more set actions:

- Set the next hop address:

  set ip next-hop *ip_address*

  You can configure multiple next-hop IP addresses in which case they are evaluated in the specified order until a valid routable next-hop IP address is found. The configured next-hops should be directly connected; otherwise the set action will not be applied.

- Set the default next hop address:

  set ip default next-hop *ip_address*

  If the normal route lookup fails for matching traffic, then the ASA forwards the traffic using this specified next-hop IP address.

- Set a recursive next hop IPv4 address:

  set ip next-hop recursive *ip_address*

  Both **set ip next-hop** and **set ip default next-hop** require that the next-hop be found on a directly connected subnet. With **set ip next-hop recursive**, the next-hop address does not need to be directly connected. Instead a recursive lookup is performed on the next-hop address, and matching traffic is forwarded to the next-hop used by that route entry according to the routing path in use on the router.

- Verify if the next IPv4 hops of a route map are available:

  set ip next-hop verify-availability *next-hop-address sequence_number* **track** *object*

  You can configure an SLA monitor tracking object to verify the reachability of the next-hop. To verify the availability of multiple next-hops, multiple **set ip next-hop verify-availability** commands can be configured with different sequence numbers and different tracking objects.

- Set the output interface for the packet:

  set interface *interface_name*

  or

  set interface null0

  This command configures the interface through which the matching traffic is forwarded. You can configure multiple interfaces, in which case they are evaluated in the specified order until a valid interface is found. When you specify **null0**, all traffic matching the route-map will be dropped. There must be a route for the destination that can be routed through the specified interface (either static or dynamic).

- Set the default interface to null0:

  set default interface null0

  If a normal route lookup fails, the ASA forwards the traffic null0, and the traffic will be dropped.

- Set the Don't Fragment (DF) bit value in the IP header:

set ip df {0|1}

- Classify IP traffic by setting a Differentiated Services Code Point (DSCP) or an IP-precedence value in the packet:

  set ip dscp *new_dscp*

Note When multiple set actions are configured, the ASA evaluates them in the following order: **set ip next-hop verify-availability; set ip next-hop**; **set ip next-hop recursive**; **set interface; set ip default next-hop**; **set default interface.**

Step 5 Configure an interface and enter interface configuration mode:

interface *interface_id*

Example:

```
ciscoasa(config)# interface GigabitEthernet0/0
```

Step 6 Configure policy based routing for through-the-box traffic:

policy-route route-map *route-map_name*

Example:

```
ciscoasa(config-if)# policy-route route-map testmap
```

To remove an existing Policy Based Routing map, simply enter the **no** form of this command.

Example:

```
ciscoasa(config-if)# no policy-route route-map testmap
```

Examples for Policy Based Routing

The following sections show examples for route map configuration, policy based routing, and a specific example of PBR in action.

Examples for Route Map Configuration

In the following example, since no action and sequence is specified, an implicit action of permit and a sequence number of 10 is assumed:

```
ciscoasa(config)# route-map testmap
```

In the following example, since no match criteria is specified, an implicit match 'any' is assumed:

```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10
```

In this example, all traffic matching <acl> will be policy routed and forwarded through outside interface.

```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl>
ciscoasa(config-route-map)# set interface outside
```

In this example, since there are no interface or next-hop actions are configured, all traffic matching <acl> will have df bit and dscp fields modified as per configuration and are forwarding using normal routing.

```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl>
set ip df 1
set ip precedence af11
```

In the following example, all traffic matching <acl_1> is forwarded using next-hop 1.1.1.10, all traffic matching <acl_2> is forwarded using next-hop 2.1.1.10 and rest of the traffic is dropped. No "match" criteria implies an implicit match "any".

```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl_1>
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10

ciscoasa(config)# route-map testmap permit 20
ciscoasa(config-route-map)# match ip address <acl_2>

ciscoasa(config-route-map)# set ip next-hop 2.1.1.10
ciscoasa(config)# route-map testmap permit 30
ciscoasa(config-route-map)# set interface Null0
```

In the following example, the route-map evaluation will be such that (i) a route-map action permit and acl action permit will apply the set actions (ii) a route-map action deny and acl action permit will skip to normal route lookup (iii) a route-map action of permit/deny and acl action deny will continue with next route-map entry. When no next route-map entry available, we will fallback to normal route lookup.

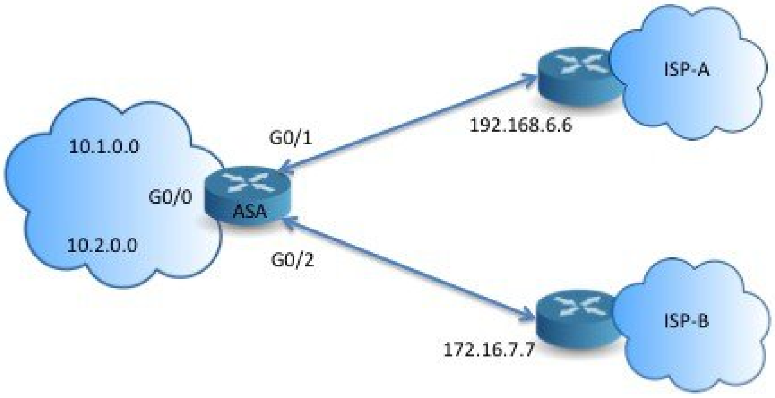
```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address permit_acl_1 deny_acl_2
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10

ciscoasa(config)# route-map testmap deny 20
ciscoasa(config-route-map)# match ip address permit_acl_3 deny_acl_4
ciscoasa(config-route-map)# set ip next-hop 2.1.1.10

ciscoasa(config)# route-map testmap permit 30
ciscoasa(config-route-map)# match ip address deny_acl_5
ciscoasa(config-route-map)# set interface outside
```

In the following example, when multiple set actions are configured, they are evaluated in the order mentioned above. Only when all options of a set action are evaluated and cannot be applied, the next set actions will be considered. This ordering will ensure that the most available and least distant next-hop will be tried first followed by next most available and least distant next-hop and so on.

```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address acl_1
ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.10 1 track 1
ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.11 2 track 2
ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.12 3 track 3
ciscoasa(config-route-map)# set ip next-hop 2.1.1.10 2.1.1.11 2.1.1.12
ciscoasa(config-route-map)# set ip next-hop recursive 3.1.1.10
ciscoasa(config-route-map)# set interface outside-1 outside-2
ciscoasa(config-route-map)# set ip default next-hop 4.1.1.10 4.1.1.11
ciscoasa(config-route-map)# set default interface Null0
```

Example Configuration for PBR

This section describes the complete set of configuration required to configure PBR for the following scenario:



First, we need to configure interfaces.

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# no shutdown
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# no shutdown
ciscoasa(config-if)# nameif outside-1
ciscoasa(config-if)# ip address 192.168.6.5 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/2
ciscoasa(config-if)# no shutdown
ciscoasa(config-if)# nameif outside-2
ciscoasa(config-if)# ip address 172.16.7.6 255.255.255.0
```

Then, we need to configure an access-list for matching the traffic.

```
ciscoasa(config)# access-list acl-1 permit ip 10.1.0.0 255.255.0.0
ciscoasa(config)# access-list acl-2 permit ip 10.2.0.0 255.255.0.0
```

We need to configure a route-map by specifying the above access-list as match criteria along with the required set actions.

```
ciscoasa(config)# route-map equal-access permit 10
ciscoasa(config-route-map)# match ip address acl-1
ciscoasa(config-route-map)# set ip next-hop 192.168.6.6

ciscoasa(config)# route-map equal-access permit 20
ciscoasa(config-route-map)# match ip address acl-2
ciscoasa(config-route-map)# set ip next-hop 172.16.7.7

ciscoasa(config)# route-map equal-access permit 30
ciscoasa(config-route-map)# set ip interface Null0
```

Now, this route-map has to be attached to an interface.

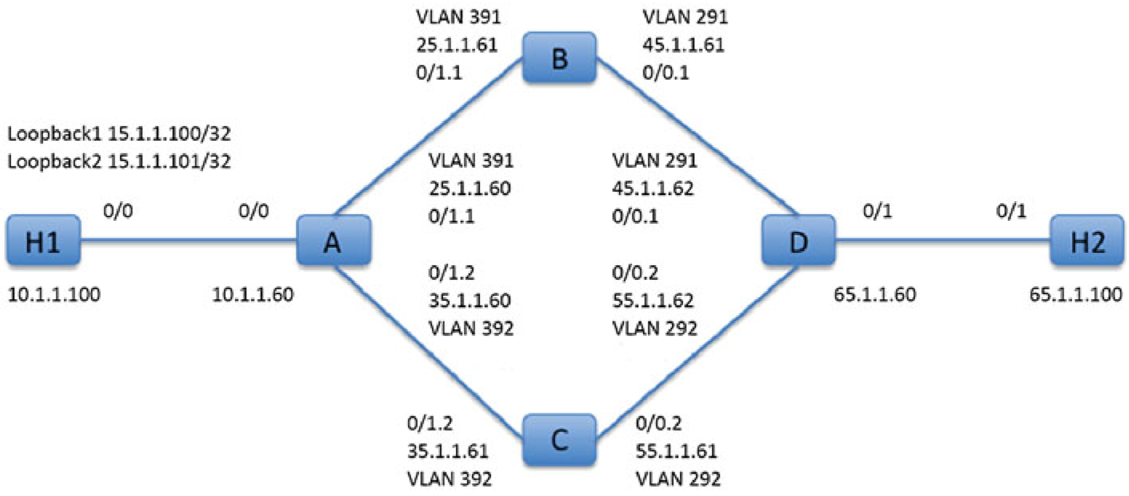
```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# policy-route route-map equal-access
```

To display the policy routing configuration.

```
ciscoasa(config)# show policy-route
Interface                   Route map
GigabitEthernet0/0   equal-access
```

Policy Based Routing in Action

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.



First, we will start with the basic configuration for all the devices involved in the set-up. Here, A, B, C, and D represent ASA devices, and H1 and H2 represent IOS routers.

ASA-A:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 10.1.1.60 255.255.255.0
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/1.1
ciscoasa(config-if)# vlan 391
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 25.1.1.60 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/1.2
ciscoasa(config-if)# vlan 392
ciscoasa(config-if)# nameif dmz
ciscoasa(config-if)# security-level 50
ciscoasa(config-if)# ip address 35.1.1.60 255.255.255.0
```

ASA-B:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/0.1
ciscoasa(config-if)# vlan 291
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 45.1.1.61 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/1.1
ciscoasa(config-if)# vlan 391
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 25.1.1.61 255.255.255.0
```

ASA-C:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/0.2
ciscoasa(config-if)# vlan 292
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 55.1.1.61 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/1.2
ciscoasa(config-if)# vlan 392
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 35.1.1.61 255.255.255.0
```

ASA-D:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# no shut
ciscoasa(config) #interface GigabitEthernet0/0.1
ciscoasa(config-if)# vlan 291
ciscoasa(config-if)# nameif inside-1
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 45.1.1.62 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/0.2
ciscoasa(config-if)# vlan 292
ciscoasa(config-if)# nameif inside-2
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 55.1.1.62 255.255.255.0

ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 65.1.1.60 255.255.255.0
```

H1:

```
ciscoasa(config)# interface Loopback1
ciscoasa(config-if)# ip address 15.1.1.100 255.255.255.255

ciscoasa(config-if)# interface Loopback2
ciscoasa(config-if)# ip address 15.1.1.101 255.255.255.255

ciscoasa(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.60
```

H2:

```
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# ip address 65.1.1.100 255.255.255.0

ciscoasa(config-if)# ip route 15.1.1.0 255.255.255.0 65.1.1.60
```

We will configure PBR on ASA-A to route traffic sourced from H1.

ASA-A:

```
ciscoasa(config-if)# access-list pbracl_1 extended permit ip host 15.1.1.100 any

ciscoasa(config-if)# route-map testmap permit 10
ciscoasa(config-if)# match ip address pbracl_1
ciscoasa(config-if)# set ip next-hop 25.1.1.61

ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# policy-route route-map testmap

ciscoasa(config-if)# debug policy-route
```

H1: ping 65.1.1.100 repeat 1 source loopback1

```
pbr: policy based route lookup called for 15.1.1.100/44397 to 65.1.1.100/0 proto 1 sub_proto
 8 received on interface inside
pbr: First matching rule from ACL(2)
pbr: route map testmap, sequence 10, permit; proceed with policy routing
pbr: evaluating next-hop 25.1.1.61
```

```
pbr: policy based routing applied; egress_ifc = outside : next_hop = 25.1.1.61
```

The packet is forwarded as expected using the next-hop address in the route-map.

When a next-hop is configured, we do a lookup in input route table to identify a connected route to the configured next-hop and use the corresponding interface. The input route table for this example is shown here (with the matching route entry highlighted).

```
in   255.255.255.255 255.255.255.255 identity
in   10.1.1.60       255.255.255.255 identity
in   25.1.1.60       255.255.255.255 identity
in   35.1.1.60       255.255.255.255 identity
in   10.127.46.17    255.255.255.255 identity
in   10.1.1.0        255.255.255.0   inside
in   25.1.1.0        255.255.255.0   outside
in   35.1.1.0        255.255.255.0   dmz
```

Next let's configure ASA-A to route packets from H1 loopback2 out of ASA-A dmz interface.

```
ciscoasa(config)# access-list pbracl_2 extended permit ip host 15.1.1.101 any

ciscoasa(config)# route-map testmap permit 20
ciscoasa(config-route-map)# match ip address pbracl
ciscoasa(config-route-map)# set ip next-hop 35.1.1.61

ciscoasa(config)# show run route-map
!
route-map testmap permit 10
  match ip address pbracl_1
  set ip next-hop 25.1.1.61
!
route-map testmap permit 20
  match ip address pbracl_2
  set ip next-hop 35.1.1.61
!
```

H1: ping 65.1.1.100 repeat 1 source loopback2

The debugs are shown here:

```
pbr: policy based route lookup called for 15.1.1.101/1234 to 65.1.1.100/1234 proto 6 sub_proto
 0 received on interface inside
pbr: First matching rule from ACL(3)
pbr: route map testmap, sequence 20, permit; proceed with policy routing
pbr: evaluating next-hop 35.1.1.61
pbr: policy based routing applied; egress_ifc = dmz : next_hop = 35.1.1.61
```

and the route entry chosen from input route table is shown here:

```
in   255.255.255.255 255.255.255.255 identity
 in   10.1.1.60       255.255.255.255 identity
 in   25.1.1.60       255.255.255.255 identity
 in   35.1.1.60       255.255.255.255 identity
 in   10.127.46.17    255.255.255.255 identity
 in   10.1.1.0        255.255.255.0   inside
 in   25.1.1.0        255.255.255.0   outside
 in   35.1.1.0        255.255.255.0   dmz
```

History for Policy Based Routing

Table 26: History for Route Maps

Feature Name	Platform Releases	Feature Information
Policy based routing	9.4(1)	Policy Based Routing (PBR) is a mechanism by which traffic is routed through specific paths with a specified QoS using ACLs. ACLs let traffic be classified based on the content of the packet's Layer 3 and Layer 4 headers. This solution lets administrators provide QoS to differentiated traffic, distribute interactive and batch traffic among low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths, and allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections. We introduced the following commands: **set ip next-hop verify-availability**, **set ip next-hop**, **set ip next-hop recursive**, **set interface**, **set ip default next-hop, set default interface**, **set ip df**, **set ip dscp**, **policy-route route-map**, **show policy-route**, **debug policy-route**

CHAPTER 25

Route Maps

This chapter describes how to configure and customize route-maps, for Cisco ASA.

About Route Maps

Route maps are used when redistributing routes into an OSPF, RIP, EIGRP or BGP routing process. They are also used when generating a default route into an OSPF routing process. A route map defines which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process.

Route maps have many features in common with widely known ACLs. These are some of the traits common to both:

- They are an ordered sequence of individual statements, and each has a permit or deny result. Evaluation of an ACL or a route map consists of a list scan, in a predetermined order, and an evaluation of the criteria of each statement that matches. A list scan is aborted once the first statement match is found and an action associated with the statement match is performed.
- They are generic mechanisms. Criteria matches and match interpretation are dictated by the way that they are applied and the feature that uses them. The same route map applied to different features might be interpreted differently.

These are some of the differences between route maps and ACLs:

- Route maps are more flexible than ACLs and can verify routes based on criteria which ACLs can not verify. For example, a route map can verify if the type of route is internal.
- Each ACL ends with an implicit deny statement, by design convention. If the end of a route map is reached during matching attempts, the result depends on the specific application of the route map. Route maps that are applied to *redistribution* behave the same way as ACLs: if the route does not match any clause in a route map then the route redistribution is denied, as if the route map contained a deny statement at the end.

Permit and Deny Clauses

Route maps can have permit and deny clauses. The deny clause rejects route matches from redistribution. You can use an ACL as the matching criterion in the route map. Because ACLs also have permit and deny clauses, the following rules apply when a packet matches the ACL:

- ACL permit + route map permit: routes are redistributed.
- ACL permit + route map deny: routes are not redistributed.
- ACL deny + route map permit or deny: the route map clause is not matched, and the next route-map clause is evaluated.

Match and Set Clause Values

Each route map clause has two types of values:

- A match value selects routes to which this clause should be applied.
- A set value modifies information that will be redistributed into the target protocol.

For each route that is being redistributed, the router first evaluates the match criteria of a clause in the route map. If the match criteria succeeds, then the route is redistributed or rejected as dictated by the permit or deny clause, and some of its attributes might be modified by the values set from the set commands. If the match criteria fail, then this clause is not applicable to the route, and the software proceeds to evaluate the route against the next clause in the route map. Scanning of the route map continues until a clause is found that matches the route or until the end of the route map is reached.

A match or set value in each clause can be missed or repeated several times, if one of these conditions exists:

- If several match entries are present in a clause, all must succeed for a given route in order for that route to match the clause (in other words, the logical AND algorithm is applied for multiple match commands).
- If a match entry refers to several objects in one entry, either of them should match (the logical OR algorithm is applied).
- If a match entry is not present, all routes match the clause.
- If a set entry is not present in a route map permit clause, then the route is redistributed without modification of its current attributes.

Note Do not configure a set entry in a route map deny clause because the deny clause prohibits route redistribution—there is no information to modify.

A route map clause without a match or set entry does perform an action. An empty permit clause allows a redistribution of the remaining routes without modification. An empty deny clause does not allow a redistribution of other routes (this is the default action if a route map is completely scanned, but no explicit match is found).

Guidelines for Route Maps

Firewall Mode

Supported only in routed firewall mode. Transparent firewall mode is not supported.

Additional Guidelines

Route maps do not support ACLs that include a user, user group, or fully qualified domain name objects.

Define a Route Map

You must define a route map when specifying which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process.

Procedure

Create the route map entry:

route-map *name* {**permit** | **deny**} [*sequence_number*]

Example:

```
ciscoasa(config)# route-map name {permit} [12]
```

Route map entries are read in order. You can identify the order using the *sequence_number* argument, or the ASA uses the order in which you add route map entries.

Customize a Route Map

This section describes how to customize the route map.

Define a Route to Match a Specific Destination Address

Procedure

Step 1 Create the route map entry:

route-map *name* {**permit** | **deny**} [*sequence_number*]

Example:

```
ciscoasa(config)# route-map name {permit} [12]
```

Route map entries are read in order. You can identify the order using the *sequence_number* option, or the ASA uses the order in which you add route map entries.

Step 2 Match any routes that have a destination network that matches a standard ACL or prefix list:

match ip address *acl_id* [*acl_id*] [...] [prefix-list]

Example:

```
ciscoasa(config-route-map)# match ip address acl1
```

If you specify more than one ACL, then the route can match any of the ACLs.

Step 3 Match any routes that have a specified metric:

match metric *metric_value*

Example:

```
ciscoasa(config-route-map)# match metric 200
```

The *metric_value* can range from 0 to 4294967295.

Step 4 Match any routes that have a next hop router address that matches a standard ACL:

match ip next-hop *acl_id* [*acl_id*] [...]

Example:

```
ciscoasa(config-route-map)# match ip next-hop acl2
```

If you specify more than one ACL, then the route can match any of the ACLs.

Step 5 Match any routes with the specified next hop interface:

match interface *if_name*

Example:

```
ciscoasa(config-route-map)# match interface if_name
```

If you specify more than one interface, then the route can match either interface.

Step 6 Match any routes that have been advertised by routers that match a standard ACL:

match ip route-source *acl_id* [*acl_id*] [...]

Example:

```
ciscoasa(config-route-map)# match ip route-source acl_id [acl_id] [...]
```

If you specify more than one ACL, then the route can match any of the ACLs.

Step 7 Match the route type:

match route-type {**internal** | **external** [**type-1** | **type-2**]}

Configure the Metric Values for a Route Action

If a route matches the **match** commands, then the following **set** commands determine the action to perform on the route before redistributing it.

To configure the metric value for a route action, perform the following steps:

Procedure

Step 1 Create the route map entry:

route-map *name* {**permit** | **deny**} [*sequence_number*]

Example:

```
ciscoasa(config)# route-map name {permit} [12]
```

Route map entries are read in order. You can identify the order using the *sequence_number* argument, or the ASA uses the order in which you add route map entries.

Step 2 Set the metric value for the route map:

set metric *metric_value*

Example:

```
ciscoasa(config-route-map)# set metric 200
```

The *metric_value* argument can range from 0 to 294967295.

Step 3 Set the metric type for the route map:

set metric-type {**type-1** | **type-2**}

Example:

```
ciscoasa(config-route-map)# set metric-type type-2
```

The *metric-type* argument can be type-1 or type-2.

Example for Route Maps

The following example shows how to redistribute routes with a hop count equal to 1 into OSPF.

The ASA redistributes these routes as external LSAs with a metric of 5 and a metric type of Type 1.

```
ciscoasa(config)# route-map 1-to-2 permit
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
```

The following example shows how to redistribute the 10.1.1.0 static route into eigrp process 1 with the configured metric value:

```
ciscoasa(config)# route outside 10.1.1.0 255.255.255.0 192.168.1.1
ciscoasa(config-route-map)# access-list mymap2 line 1 permit 10.1.1.0 255.255.255.0
ciscoasa(config-route-map)# route-map mymap2 permit 10
ciscoasa(config-route-map)# match ip address mymap2
ciscoasa(config-route-map)# router eigrp 1
ciscoasa(config-router)# redistribute static metric 250 250 1 1 1 route-map mymap2
```

History for Route Maps

Table 27: Feature History for Route Maps

Feature Name	Platform Releases	Feature Information
Route maps	7.0(1)	We introduced this feature. We introduced the following command: **route-map**.
Enhanced support for static and dynamic route maps	8.0(2)	Enhanced support for dynamic and static route maps was added.
Support for Stateful Failover of dynamic routing protocols (EIGRP, OSPF, and RIP) and debugging of general routing-related operations	8.4(1)	We introduced the following commands: **debug route**, **show debug route**. We modified the following command: **show route**.
Dynamic Routing in Multiple Context Mode	9.0(1)	Route maps are supported in multiple context mode.
Support for BGP	9.2(1)	We introduced this feature. We introduced the following commands: **router bgp**
IPv6 support for Prefix Rule	9.3.2	We introduced this feature.

CHAPTER 26

BGP

This chapter describes how to configure the Cisco ASA to route data, perform authentication, and redistribute routing information using the Border Gateway Protocol (BGP).

About BGP

BGP is an inter and intra autonomous system routing protocol. An autonomous system is a network or group of networks under a common administration and with common routing policies. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISP).

When to Use BGP

Customer networks, such as universities and corporations, usually employ an Interior Gateway Protocol (IGP) such as OSPF for the exchange of routing information within their networks. Customers connect to ISPs, and ISPs use BGP to exchange customer and ISP routes. When BGP is used between autonomous systems (AS), the protocol is referred to as External BGP (EBGP). If a service provider is using BGP to exchange routes within an AS, then the protocol is referred to as Interior BGP (IBGP).

BGP can also be used for carrying routing information for IPv6 prefix over IPv6 networks.

Note When a BGPv6 device joins the cluster, it generates a soft traceback when logging level 7 is enabled.

Routing Table Changes

BGP neighbors exchange full routing information when the TCP connection between neighbors is first established. When changes to the routing table are detected, the BGP routers send to their neighbors only those routes that have changed. BGP routers do not send periodic routing updates, and BGP routing updates advertise only the optimal path to a destination network.

Note AS loop detection is done by scanning the full AS path (as specified in the AS_PATH attribute), and checking that the AS number of the local system does not appear in the AS path. By default, EBGP advertises the learned routes to the same peer to prevent additional CPU cycles on the ASA in performing loop checks and to avoid delays in the existing outgoing update tasks.

Routes learned via BGP have properties that are used to determine the best route to a destination, when multiple paths exist to a particular destination. These properties are referred to as BGP attributes and are used in the route selection process:

- Weight—This is a Cisco-defined attribute that is local to a router. The weight attribute is not advertised to neighboring routers. If the router learns about more than one route to the same destination, the route with the highest weight is preferred.
- Local preference—The local preference attribute is used to select an exit point from the local AS. Unlike the weight attribute, the local preference attribute is propagated throughout the local AS. If there are multiple exit points from the AS, the exit point with the highest local preference attribute is used as an exit point for a specific route.
- Multi-exit discriminator—The multi-exit discriminator (MED) or metric attribute is used as a suggestion to an external AS regarding the preferred route into the AS that is advertising the metric. It is referred to as a suggestion because the external AS that is receiving the MEDs may also be using other BGP attributes for route selection. The route with the lower MED metric is preferred.
- Origin—The origin attribute indicates how BGP learned about a particular route. The origin attribute can have one of three possible values and is used in route selection.
    - IGP—The route is interior to the originating AS. This value is set when the network router configuration command is used to inject the route into BGP.
    - EGP—The route is learned via the Exterior Border Gateway Protocol (EBGP).
    - Incomplete—The origin of the route is unknown or learned in some other way. An origin of incomplete occurs when a route is redistributed into BGP.
- AS_path—When a route advertisement passes through an autonomous system, the AS number is added to an ordered list of AS numbers that the route advertisement has traversed. Only the route with the shortest AS_path list is installed in the IP routing table.
- Next hop—The EBGP next-hop attribute is the IP address that is used to reach the advertising router. For EBGP peers, the next-hop address is the IP address of the connection between the peers. For IBGP, the EBGP next-hop address is carried into the local AS.
- Community—The community attribute provides a way of grouping destinations, called communities, to which routing decisions (such as acceptance, preference, and redistribution) can be applied. Route maps are used to set the community attribute. The predefined community attributes are as follows:
    - no-export—Do not advertise this route to EBGP peers.
    - no-advertise—Do not advertise this route to any peer.
    - internet—Advertise this route to the Internet community; all routers in the network belong to it.

BGP Path Selection

BGP may receive multiple advertisements for the same route from different sources. BGP selects only one path as the best path. When this path is selected, BGP puts the selected path in the IP routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order presented, to select a path for a destination:

- If the path specifies a next hop that is inaccessible, drop the update.
- Prefer the path with the largest weight.
- If the weights are the same, prefer the path with the largest local preference.
- If the local preferences are the same, prefer the path that was originated by BGP running on this router.
- If no route was originated, prefer the route that has the shortest AS_path.
- If all paths have the same AS_path length, prefer the path with the lowest origin type (where IGP is lower than EGP, and EGP is lower than incomplete).
- If the origin codes are the same, prefer the path with the lowest MED attribute.
- If the paths have the same MED, prefer the external path over the internal path.
- If the paths are still the same, prefer the path through the closest IGP neighbor.
- Determine if multiple paths require installation in the routing table for BGP Multipath, on page 639.
- If both paths are external, prefer the path that was received first (the oldest one).
- Prefer the path with the lowest IP address, as specified by the BGP router ID.
- If the originator or router ID is the same for multiple paths, prefer the path with the minimum cluster list length.
- Prefer the path that comes from the lowest neighbor address.

BGP Multipath

BGP Multipath allows installation into the IP routing table of multiple equal-cost BGP paths to the same destination prefix. Traffic to the destination prefix is then shared across all installed paths.

These paths are installed in the table together with the best path for load-sharing. BGP Multipath does not affect best-path selection. For example, a router still designates one of the paths as the best path, according to the algorithm, and advertises this best path to its BGP peers.

In order to be candidates for multipath, paths to the same destination need to have these characteristics equal to the best-path characteristics:

- Weight
- Local preference
- AS-PATH length
- Origin code
- Multi Exit Discriminator (MED)
- One of these:

  - Neighboring AS or sub-AS (before the addition of the BGP Multipaths)
  - AS-PATH (after the addition of the BGP Multipaths)

Some BGP Multipath features put additional requirements on multipath candidates:

- The path should be learned from an external or confederation-external neighbor (eBGP).
- The IGP metric to the BGP next hop should be equal to the best-path IGP metric.

These are the additional requirements for internal BGP (iBGP) multipath candidates:

- The path should be learned from an internal neighbor (iBGP).
- The IGP metric to the BGP next hop should be equal to the best-path IGP metric, unless the router is configured for unequal-cost iBGP multipath.

BGP inserts up to *n* most recently received paths from multipath candidates into the IP routing table, where *n* is the number of routes to install to the routing table, as specified when you configure BGP Multipath. The default value, when multipath is disabled, is 1.

For unequal-cost load balancing, you can also use BGP Link Bandwidth.

Note The equivalent next-hop-self is performed on the best path that is selected among eBGP multipaths before it is forwarded to internal peers.

Guidelines for BGP

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Does not support transparent firewall mode. BGP is supported only in routed mode.

IPv6 Guidelines

Supports IPv6. Graceful restart is not supported for IPv6 address family.

Additional Guidelines

- The system does not add route entry for the IP address received over PPPoE in the CP route table. BGP always looks into CP route table for initiating the TCP session, hence BGP does not form TCP session.

  Thus, BGP over PPPoE is not supported.

- To avoid adjacency flaps due to route updates being dropped if the route update is larger than the minimum MTU on the link, ensure that you configure the same MTU on the interfaces on both sides of the link.

Configure BGP

This section describes how to enable and configure the BGP process on your system.

Procedure

Step 1 Enable BGP, on page 641.

Step 2 Define the Best Path for a BGP Routing Process, on page 643.

Step 3 Configure Policy Lists, on page 643.

Step 4 Configure AS Path Filters, on page 645.

Step 5 Configure Community Rules, on page 645.

Step 6 Configure IPv4 Address Family Settings, on page 646.

Step 7 Configure IPv6 Address Family Settings, on page 659.

Enable BGP

This section describes the steps required to enable BGP routing, establish a BGP routing process and configure general BGP parameters.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Valid values for autonomous-num are from 1-4294967295 and 1.0-XX.YY.

Step 2 Discard routes that have as-path segments that exceed a specified value:

bgp maxas-limit number

Example:

```
ciscoasa(config-router)# bgp maxas-limit 15
```

The number argument specifies the maximum number of autonomous system segments, allowed. Valid values are from 1 to 254.

Step 3 Log BGP neighbor resets:

bgp log-neighbor-changes

Step 4 Enable BGP to automatically discover the best TCP path MTU for each BGP session:

bgp transport path-mtu-discovery

Step 5 Enable BGP to terminate external BGP sessions of any directly adjacent peer if the link used to reach the peer goes down; without waiting for the hold-down timer to expire:

bgp fast-external-fallover

Step 6 Allow a BGP routing process to discard updates received from an external BGP (eBGP) peers that do not list their autonomous system (AS) number as the first AS path segment in the AS_PATH attribute of the incoming route:

bgp enforce-first-as

Step 7 Change the default display and regular expression match format of BGP 4-byte autonomous system numbers from asplain (decimal values) to dot notation:

bgp asnotation dot

Step 8 Adjust BGP network timers:

timers bgp keepalive holdtime [min-holdtime]

Example:

```
ciscoasa(config-router)# timers bgp 80 120
```

- keepalive — frequency (in seconds) with which the ASA sends keepalive messages to its peer. The default value 60 seconds.
- holdtime — interval (in seconds) after not receiving a keepalive message that the ASA declares a peer dead. The default is 180 seconds.
- (Optional) min-holdtime — interval (in seconds) after not receiving a keepalive message from a neighbor, that the ASA declares a neighbor dead.

Step 9 Enable BGP graceful restart capability:

bgp graceful-restart [restart-time seconds|stalepath-time seconds][all]

Example:

```
ciscoasa(config-router)# bgp graceful-restart restart-time 200
```

- restart-time — maximum time period (in seconds) that the ASA will wait for a graceful-restart-capable neighbor to return to normal operation after a restart event occurs. The default is 120 seconds. Valid values are from 1 to 3600 seconds.
- stalepath-time — maximum time period (in seconds) that the ASA will hold stale paths for a restarting peer. All stale paths are deleted after this timer expires. The default value is 360 seconds. Valid values are from 1 to 3600 seconds.

Define the Best Path for a BGP Routing Process

This section describes the steps required to configure the BGP best path. For more information on the best path, see BGP Path Selection, on page 639.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Change the default local preference value:

bgp default local-preference number

Example:

```
ciscoasa(config-router)# bgp default local-preference 500
```

The number argument is any value between 0 and 4294967295. Higher values indicate higher preference.

The default value is 100.

Step 3 Enable Multi Exit Discriminator (MED) comparison among paths learned from neighbors in different autonomous systems:

bgp always-compare-med

Step 4 Compare between similar routes received from external BGP (eBGP) peers during the best path selection process and switch the best path to the route with the lowest router ID:

bgp bestpath compare-routerid

Step 5 Select the best MED path advertised from the neighboring AS:

bgp deterministic-med

Step 6 Set a path with a missing MED attribute as the least preferred path:

bgp bestpath med missing-as-worst

Configure Policy Lists

When a policy list is referenced within a route map, all of the match statements within the policy list are evaluated and processed. Two or more policy lists can be configured with a route map. A policy list can also coexist with any other preexisting match and set statements that are configured within the same route map but outside of the policy list. This section describes the steps required to configure policy lists.

Procedure

Step 1 Create a BGP policy list.

policy-list *policy_list_name* {**permit** | **deny**}

The **permit** keyword allows access for matching conditions.

The **deny** keyword denies access for matching conditions.

Example:

```
ciscoasa(config)# policy-list Example-policy-list1 permit
```

Step 2 Distribute routes that have their next hop out of one of the interfaces specified:

match interface [*interface_name* [*interface_name*] […]]

Example:

```
ciscoasa(config-policy-list)# match interface outside
```

Step 3 Redistribute routes by matching either or all of the following: the destination address, next hop router address, and router/access server source:

match ip {**address** | **next-hop** | **route-source**}

Step 4 Match a BGP autonomous system path:

match as-path

Step 5 Match a BGP community:

match community {*community-list_name* | **exact-match**}

- *community-list_name* — one or more community lists.
- **exact-match** — indicates that an exact match is required. All of the communities and only those communities specified must be present.

Example:

```
ciscoasa(config-policy-list)# match community ExampleCommunity1
```

Step 6 Redistribute routes with the metrics specified:

match metric *metric* [*metric* […]]

Step 7 Redistribute routes in the routing table that match the specified tags:

match tag *tag* [*tag* […]]

Configure AS Path Filters

An AS path filter allows you to filter the routing update message by using access lists and look at the individual prefixes within an update message. If a prefix within the update message matches the filter criteria then that individual prefix is filtered out or accepted depending on what action the filter entry has been configured to carry out. This section describes the steps required to configure AS path filters.

Note The as-path access-lists are not the same as the regular firewall ACLs.

Procedure

Configure an autonomous system path filter using a regular expression in the global configuration mode:

as-path access-list acl-number {permit|deny} regexp

Example:

```
ciscoasa(config)# as-path access-list 35 permit testaspath
```

- *acl-number* — AS-path access-list number. Valid values are from 1 to 500.
- regexp — regular expression that defines the AS-path filter. The autonomous system number is expressed in the range from 1 to 65535.

Configure Community Rules

A community is a group of destinations that share some common attribute. You can use community lists to create groups of communities to use in a match clause of a route map. Just like an access list, a series of community lists can be created. Statements are checked until a match is found. As soon as one statement is satisfied, the test is concluded. This section describes the steps required to configure community rules.

Procedure

Create or configure a BGP community list and control access to it:

community-list {standard| community list-name {deny|permit} [community-number] [AA:NN] [internet] [no-advertise][no-export]}| {expanded|expanded list-name {deny| permit}regexp}

Example:

```
ciscoasa(config)# community-list standard excomm1 permit 100 internet no-advertise no-export
```

- standard — configures a standard community list using a number from 1 to 99 to identify one or more permit or deny groups of communities.

- (Optional) community-number — community as a 32-bit number from 1 to 4294967200. A single community can be entered or multiple communities can be entered, each separated by a space.
- AA:NN — an autonomous system number and network number entered in the 4-byte new community format. This value is configured with two 2-byte numbers separated by a colon. A number from 1 to 65535 can be entered for each 2-byte number. A single community can be entered or multiple communities can be entered, each separated by a space.
- (Optional) internet — specifies the Internet community. Routes with this community are advertised to all peers (internal and external).
- (Optional) no-advertise — specifies the no-advertise community. Routes with this community are not advertised to any peer (internal or external).
- (Optional) no-export — specifies the no-export community. Routes with this community are advertised to only peers in the same autonomous system or to only other subautonomous systems within a confederation. These routes are not advertised to external peers.
- (Optional) expanded— configures an expanded community list number from 100 to 500 to identify one or more permit or deny groups of communities.
- regexp — regular expression that defines the AS-path filter. The autonomous system number is expressed in the range from 1 to 65535.

  Note Regular expressions can be used only with expanded community lists.

Configure IPv4 Address Family Settings

The IPv4 settings for BGP can be set up from the IPv4 family option within the BGP configuration setup. The IPv4 family section includes subsections for General settings, Aggregate address settings, Filtering settings and Neighbor settings. Each of these subsections enable you to customize parameters specific to the IPv4 family.

Configure IPv4 Family General Settings

This section describes the steps required to configure the general IPv4 settings.

Procedure

Step 1 Enable a BGP routing process, which places the router in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 (Optional) Configure a fixed router ID for the local BGP routing process:

bgp router-id A.B.C.D

Example:

```
ciscoasa(config-router-af)# bgp router-id 10.86.118.3
```

The argument A.B.C.D specifies a router identifier in the form of an IP address. If you do not specify a router ID, it is automatically assigned.

Step 4 (Optional) Configure a cluster pool of IP addresses in the Individual Interface (L3) mode:

bgp router-id cluster-pool

Example:

```
ciscoasa(config-router-af)# bgp router-id cp
```

Note In an L3 cluster, you cannot define a BGP neighbor as one of the cluster pool IP addresses.

Step 5 Configure the administrative distance for BGP routes:

distance bgp external-distance internal-distance local-distance

Example:

```
ciscoasa(config-router-af)# distance bgp 80 180 180
```

- external-distance — administrative distance for external BGP routes. Routes are external when learned from an external autonomous system. The range of values for this argument are from 1 to 255.
- internal-distance — administrative distance for internal BGP routes. Routes are internal when learned from peer in the local autonomous system. The range of values for this argument are from 1 to 255.
- local-distance — administrative distance for local BGP routes. Local routes are those networks listed with a network router configuration command, often as back doors, for the router or for the networks that is being redistributed from another process. The range of values for this argument are from 1 to 255.

Step 6 Modify metric and tag values when the IP routing table is updated with BGP learned routes:

table-map {WORD|route-map_name}

Example:

```
ciscoasa(config-router-af)# table-map example1
```

The argument route-map_name specifies the route map name from the route-map command.

Step 7 Configure a BGP routing process to distribute a default route (network 0.0.0.0):

default-information originate

Step 8 Configure automatic summarization of subnet routes into network-level routes:

auto-summary

Step 9 Suppress the advertisement of routes that are not installed in the routing information base (RIB):

bgp suppress-inactive

Step 10 Synchronize between BGP and your Interior Gateway Protocol (IGP) system:

synchronization

Step 11 Configure iBGP redistribution into an IGP, such as OSPF:

bgp redistribute-internal

Step 12 Configure scanning intervals of BGP routers for next hop validation:

bgp scan-time scanner-interval

Example:

```
ciscoasa(config-router-af)# bgp scan-time 15
```

The argument scanner-interval specifies scanning interval of BGP routing information. Valid values are from 5 to 60 seconds. The default is 60 seconds.

Step 13 Configure BGP next-hop address tracking:

bgp nexthop trigger {delay seconds|enable}

Example:

```
ciscoasa(config-router-af)# bgp nexthop trigger delay 15
```

- trigger — specifies the use of BGP next-hop address tracking. Use this keyword with the delay keyword to change the next-hop tracking delay. Use this keyword with the enable keyword to enable next-hop address tracking.
- delay — changes the delay interval between checks on updated next-hop routes installed in the routing table.
- seconds — specifies the delay in seconds. Range is from 0 to 100. Default is 5.
- enable — enables BGP next-hop address tracking immediately.

Step 14 Control the maximum number of parallel iBGP routes that can be installed in a routing table:

maximum-paths {number_of_paths|ibgp number_of_paths}

Example:

```
ciscoasa(config-router-af)# maximum-paths ibgp 2
```

Note If the ibgp keyword is not used, then the number_of_paths argument controls the maximum number of parallel EBGP routes.

The number_of_paths argument specifies the number of routes to install to the routing table. Valid values are between 1 and 8.

Configure IPv4 Family Aggregate Address Settings

This section describes the steps required to define the aggregation of specific routes into one route.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Create an aggregate entry in a BGP database:

aggregate-address address mask [as-set][summary-only][suppress-map map-name][advertise-map map-name][attribute-map map-name]

Example:

```
ciscoasa(config-router-af) aggregate-address 10.86.118.0 255.255.255.0 as-set summary-only
 suppress-map example1 advertise-map example1 attribute-map example1
```

- address — the aggregate address.
- mask — the aggregate mask.
- map-name — the route map.
- (Optional) as-set — generates autonomous system set path information.
- (Optional) summary-only — filters all more-specific routes from updates.
- (Optional) Suppress-map map-name —specifies the name of the route map used to select the routes to be suppressed.
- (Optional) Advertise-map map-name — specifies the name of the route map used to select the routes to create AS_SET origin communities.

- (Optional) Attribute-map map-name — specifies the name of the route map used to set the attribute of the aggregate route.

Configure IPv4 Family Filtering Settings

This section describes the steps required to filter routes or networks received in incoming BGP updates.

Procedure

Step 1 Enable a BGP routing process and enter router configuration mode:

router bgp *autonomous-num*

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [**unicast**]

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Filter routes or networks received in incoming or advertised in outgoing BGP updates:

distribute-list *acl-number* {**in** | **out**} [*protocol process-number* | **connected** | **static**]

The argument *acl-number* specifies IP access list number. The access list defines which networks are to be received and which are to be suppressed in routing updates.

The keyword **in** specifies that the filter must be applied to incoming BGP updates and **out** specifies that the filter must be applied to outgoing BGP updates.

For outbound filters, you can optionally specify a protocol (**bgp**, **eigrp**, **ospf**, or **rip**) with a process number (except for RIP) to apply to the distribution list. You can also filter on whether the peers and networks were learned through **connected** or **static** routes.

Example:

```
ciscoasa(config-router-af)# distribute-list ExampleAcl in bgp 2
```

Configure IPv4 Family BGP Neighbor Settings

This section describes the steps required to define BGP neighbors and neighbor settings.

Procedure

Step 1 Enable a BGP routing process, which places the router in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Add an entry to the BGP neighbor table:

neighbor ip-address remote-as autonomous-number

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 remote-as 3
```

Step 4 (Optional) Disable a neighbor or peer group:

neighbor ip-address shutdown

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 shutdown 3
```

Step 5 Exchange information with a BGP neighbor:

neighbor ip-address activate

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 activate
```

Step 6 Enable or disable the Border Gateway Protocol (BGP) graceful restart capability for a BGP neighbor:

neighbor ip-address ha-mode graceful-restart [disable]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 ha-mode graceful-restart
```

(Optional) The disable keyword disables BGP graceful restart capability for a neighbor.

Step 7 Distribute BGP neighbor information as specified in an access list:

neighbor {ip-address} distribute-list {access-list-name}{in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 distribute-list ExampleAcl in
```

- access-list-number — the number of a standard or extended access list. The range of a standard access list number is from 1 to 99. The range of an extended access list number is from 100 to 199.
- expanded-list-number — the number of an expanded access list number. The range of an expanded access list is from 1300 to 2699.
- access-list-name — the name of a standard or extended access list.
- prefix-list-name — the name of a BGP prefix list.
- in — the access list is applied to incoming advertisements to that neighbor.
- out — that the access list is applied to outgoing advertisements to that neighbor.

Step 8 Apply a route map to incoming or outgoing routes:

neighbor {ip-address} route-map map-name {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 route-map example1 in
```

The keyword in applies a route map to incoming routes.

The keyword out applies a route map to outgoing routes.

Step 9 Distribute BGP neighbor information as specified in a prefix list:

neighbor {ip-address} prefix-list prefix-list-name {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 prefix-list NewPrefixList in
```

The keyword in implies that the prefix list is applied to incoming advertisements from that neighbor.

The keyword out implies that the prefix list is applied to outgoing advertisements to that neighbor.

Step 10 Set up a filter list:

neighbor {ip-address} filter-list access-list-number {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 filter-list 5 in
```

- access-list-name — specifies the number of an autonomous system path access list. You define this access list with the ip as-path access-list command.
- in — that the access list is applied to incoming advertisements from that neighbor.
- out — that the access list is applied to outgoing advertisements to that neighbor.

Step 11 Control the number of prefixes that can be received from a neighbor:

neighbor {ip-address} maximum-prefix maximum [threshold][restart restart interval][warning-only]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 maximum-prefix 7 75 restart 12
```

- maximum — the maximum number of prefixes allowed from this neighbor.
- (Optional) threshold — integer specifying at what percentage of maximum the router starts to generate a warning message. The range is from 1 to 100; the default is 75 (percent).
- (Optional) restart interval — integer value (in minutes) that specifies the time interval after which the BGP neighbor restarts.
- (Optional) warning-only — allows the router to generate a log message when the maximum number of prefixes is exceeded, instead of terminating the peering.

Step 12 Allow a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route:

neighbor {ip-address} default-originate [route-map map-name]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 default-originate route-map example1
```

The argument map-name is the name of the route-map. The route map allows route 0.0.0.0 to be injected conditionally.

Step 13 Set the minimum interval between the sending of BGP routing updates:

neighbor {ip-address} advertisement-interval seconds

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 advertisement-interval 15
```

The argument seconds is the time (in seconds). Valid values are from 0 to 600.

Step 14 Advertise the routes in the BGP table that matches the configured route-map:

neighbor {ip-address} advertise-map map-name {exist-map map-name |non-exist-map map-name}[check-all-paths]

Example:

```
ciscoasa(config-router-af)# neighbor 10.2.1.1 advertise-map MAP1 exist-map MAP2
```

- advertise-map map name — the name of the route map that will be advertised if the conditions of the exist map or non-exist map are met.
- exist-map map name — the name of the exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not.
- non-exist-map map name — the name of the non-exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not.

- (Optional) check all paths — enables checking of all paths by the exist-map with a prefix in the BGP table.

Step 15 Remove private autonomous system numbers from outbound routing updates:

neighbor {ip-address} remove-private-as

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 remove-private-as
```

Step 16 Sets the timers for a specific BGP peer or peer group.

neighbor {ip-address} timers keepalive holdtime min holdtime

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 timers 15 20 12
```

- keepalive — the frequency (in seconds) with which the ASA sends keepalive messages to its peer. The default is 60 seconds. Valid values are from 0 to 65535.
- holdtime — the interval (in seconds) after not receiving a keepalive message that the ASA declares a peer dead. The default is 180 seconds.
- min holdtime — the minimum interval (in seconds) after not receiving a keepalive message that the ASA declares a peer dead.

Step 17 Enable Message Digest 5 (MD5) authentication on a TCP connection between two BGP peers:

neighbor {ip-address} password string

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 password test
```

The argument string is a case-sensitive password of up to 25 characters when the service password-encryption command is enabled and up to 81 characters when the service password-encryption command is not enabled. The string can contain any alphanumeric characters, including spaces.

Note When you set the first character of the password as a number, do not provide a space immediately after the number. That is, you cannot specify a password in the format number-space-anything. The space after the number can cause authentication to fail.

Step 18 Specify that communities attributes should be sent to a BGP neighbor:

neighbor {ip-address} send-community

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 send-community
```

Step 19 Configure the router as the next hop for a BGP-speaking neighbor or peer group:

neighbor {ip-address}next-hop-self

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 next-hop-self
```

Step 20 Accept and attempt BGP connections to external peers residing on networks that are not directly connected:

neighbor {ip-address} ebgp-multihop [ttl]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 ebgp-multihop 5
```

The argument ttl specifies time-to-live in the range from 1 to 255 hops.

Step 21 Disable connection verification to establish an eBGP peering session with a single-hop peer that uses a loopback interface:

neighbor {ip-address} disable-connected-check

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 disable-connected-check
```

Step 22 Secure a BGP peering session and configures the maximum number of hops that separate two external BGP (eBGP) peers:

neighbor {ip-address} ttl-security hops hop-count

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 ttl-security hops 15
```

The argument hop-count is the number of hops that separate the eBGP peers. The TTL value is calculated by the router from the configured hop-count argument. Valid values are from 1 to 254.

Step 23 Assign a weight to a neighbor connection:

neighbor {ip-address} weight number

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 weight 30
```

The argument number is the weight to assign to a neighbor connection. Valid values are from 0 to 65535.

Step 24 Configure the ASA to accept only a particular BGP version:

neighbor {ip-address} version number

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 version 4
```

The argument number specifies the BGP version number. The version can be set to 2 to force the software to use only Version 2 with the specified neighbor. The default is to use Version 4 and dynamically negotiate down to Version 2 if requested.

Step 25 Enable a TCP transport session option for a BGP session:

neighbor {ip-address} transport {connection-mode{active|passive}| path-mtu-discovery[disable]}

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 transport path-mtu-discovery
```

- connection-mode — the type of connection (active or passive).
- path-mtu-discovery — enables TCP transport path maximum transmission unit (MTU) discovery. TCP path MTU discovery is enabled by default.
- (Optional) disable — disables TCP path MTU discovery.

Step 26 Customize the AS_PATH attribute for routes received from an external Border Gateway Protocol (eBGP) neighbor:

neighbor {ip-address} local-as [autonomous-system-number[no-prepend]]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 local-as 5 no-prepend replace-as
```

- (Optional) autonomous-system-number — the number of an autonomous system to prepend to the AS_PATH attribute. The range of values for this argument is any valid autonomous system number from 1 to 4294967295 or 1.0 to XX.YY.
- (Optional) no-prepend — does not prepend the local autonomous system number to any routes received from the eBGP neighbor.

Configure IPv4 Network Settings

This section describes the steps required to define the networks to be advertised by the BGP routing process.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Specify the networks to be advertised by the BGP routing processes:

network {network-number [mask network-mask]}[route-map map-tag]

Example:

```
ciscoasa(config-router-af)# network 10.86.118.13 mask 255.255.255.255 route-map example1
```

- network-number — the network that BGP will advertise.
- (Optional) network-mask — the network or subnetwork mask with mask address.
- (Optional) map-tag — the identifier of a configured route map. The route map should be examined to filter the networks to be advertised. If not specified, all networks are advertised.

Configure IPv4 Redistribution Settings

This section describes the steps required to define the conditions for redistributing routes from another routing domain into BGP.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

Example:

```
ciscoasa(config-router)# address-family ipv4[unicast]
```

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Redistribute routes from another routing domain into a BGP autonomous system:

redistribute protocol [process-id] [metric] [route-map [map-tag]]

Example:

```
ciscoasa(config-router-af)# redistribute ospf 2 route-map example1 match external
```

- protocol — the source protocol from which routes are being redistributed. It can be one of the following: Connected, EIGRP, OSPF, RIP or Static.
- (Optional) process-id — a name for the specific routing process.
- (Optional) metric — the metric for the redistributed route.
- (Optional) map-tag — the identifier of a configured route map.

Note The route map should be examined to filter the networks to be redistributed. If not specified, all networks are redistributed.

Configure IPv4 Route Injection Settings

This section describes the steps required to define the routes to be conditionally injected into the BGP routing table.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes:

address-family ipv4 [unicast]

Example:

```
ciscoasa(config-router)# address-family ipv4[unicast]
```

The keyword unicast specifies IPv4 unicast address prefixes. This is the default, even if not specified.

Step 3 Configure conditional route injection to inject more specific routes into a BGP routing table:

bgp inject-map inject-map exist-map exist-map [copy-attributes]

Example:

```
ciscoasa(config-router-af)# bgp inject-map example1 exist-map example2 copy-attributes
```

- inject-map — the name of the route map that specifies the prefixes to inject into the local BGP routing table.
- exist-map — the name of the route map containing the prefixes that the BGP speaker will track.

- (Optional) copy-attributes — configures the injected route to inherit attributes of the aggregate route.

Configure IPv6 Address Family Settings

The IPv6 settings for BGP can be set up from the IPv6 family option within the BGP configuration setup. The IPv6 family section includes subsections for General settings, Aggregate address settings and Neighbor settings. Each of these subsections enable you to customize parameters specific to the IPv6 family.

This section describes how to customize the BGP IPv6 family settings.

Configure IPv6 Family General Settings

This section describes the steps required to configure the general IPv6 settings.

Procedure

Step 1 Enable a BGP routing process, which places the router in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 [unicast]

Step 3 Configure the administrative distance for BGP routes:

distance bgp external-distance internal-distance local-distance

Example:

```
ciscoasa(config-router-af)# distance bgp 80 180 180
```

- external-distance — administrative distance for external BGP routes. Routes are external when learned from an external autonomous system. The range of values for this argument are from 1 to 255.
- internal-distance — administrative distance for internal BGP routes. Routes are internal when learned from peer in the local autonomous system. The range of values for this argument are from 1 to 255.
- local-distance — administrative distance for local BGP routes. Local routes are those networks listed with a network router configuration command, often as back doors, for the router or for the networks that is being redistributed from another process. The range of values for this argument are from 1 to 255.

Step 4 (Optional) Configure a BGP routing process to distribute a default route (network 0.0.0.0):

default-information originate

Step 5 (Optional) Suppress the advertisement of routes that are not installed in the routing information base (RIB):

bgp suppress-inactive

Step 6 Synchronize between BGP and your Interior Gateway Protocol (IGP) system:

synchronization

Step 7 Configure iBGP redistribution into an IGP, such as OSPF:

bgp redistribute-internal

Step 8 Configure scanning intervals of BGP routers for next hop validation:

bgp scan-time scanner-interval

Example:

```
ciscoasa(config-router-af)# bgp scan-time 15
```

Valid values for the scanner-interval argument from 5 to 60 seconds. The default is 60 seconds.

Step 9 Control the maximum number of parallel iBGP routes that can be installed in a routing table:

maximum-paths {number_of_paths|ibgp number_of_paths}

Example:

```
ciscoasa(config-router-af)# maximum-paths ibgp 2
```

Valid values for the number_of_paths argument is between 1 and 8.

If the ibgp keyword is not used, then the number_of_paths argument controls the maximum number of parallel EBGP routes.

Configure IPv6 Family Aggregate Address Settings

This section describes the steps required to define the aggregation of specific routes into one route.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 unicast

Step 3 Create an aggregate entry in a BGP database:

aggregate-address ipv6-address/cidr [as-set][summary-only][suppress-map map-name][advertise-map ipv6-map-name][attribute-map map-name]

Example:

```
ciscoasa(config-router-af) aggregate-address 2000::1/8 summary-only
```

- address — the aggregate IPv6 address.
- (Optional) as-set — generates autonomous system set path information.
- (Optional) summary-only — filters all more-specific routes from updates.
- (Optional) suppress-map map-name — specifies the name of the route map used to select the routes to be suppressed.
- (Optional) advertise-map map-name — specifies the name of the route map used to select the routes to create AS_SET origin communities.
- (Optional) attribute-map map-name — specifies the name of the route map used to set the attribute of the aggregate route.

Step 4 Set the interval at which BGP routes will be aggregated:

bgp aggregate-timer seconds

Example:

```
ciscoasa(config-router-af)bgp aggregate-timer 20
```

Configure IPv6 Family BGP Neighbor Settings

This section describes the steps required to define BGP neighbors and neighbor settings.

Procedure

Step 1 Enable a BGP routing process, which places the router in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 [unicast]

Step 3 Add an entry to the BGP neighbor table:

neighbor ipv6-address remote-as autonomous-number

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1/8 remote-as 3
```

The argument ipv6-address specifies the IPv6 address of the next hop that can be used to reach the specified network.TheIPv6 address of the next hop need not be directly connected; recursion is done to find the IPv6 address of the directly connected next hop.When an interface type and interface number are specified, you can optionally specify the IPv6 address of the next hop to which packets are output.You must specify an interface type and an interface number when using a link-local address as the next hop (the link-local next hop must also be an adjacent device).

Note This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

Step 4 (Optional) Disable a neighbor or peer group:

neighbor ipv6-address shutdown

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1/8 shutdown 3
```

Step 5 Exchange information with a BGP neighbor:

neighbor ipv6-address activate

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1/8 activate
```

Step 6 Apply a route map to incoming or outgoing routes:

neighbor {ipv6-address} route-map map-name {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 route-map example1 in
```

The keyword in applies a route map to incoming routes.

The keyword out applies a route map to outgoing routes.

Step 7 Distribute BGP neighbor information as specified in a prefix list:

neighbor {ipv6-address} prefix-list prefix-list-name {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 prefix-list NewPrefixList in
```

The keyword in implies that the prefix list is applied to incoming advertisements from that neighbor.

The keyword out implies that the prefix list is applied to outgoing advertisements to that neighbor.

Step 8 Set up a filter list:

neighbor {ipv6-address} filter-list access-list-name {in|out}

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 filter-list 5 in
```

- access-list-name — specifies the number of an autonomous system path access list. You define this access list with the ip as-path access-list command.
- in — that the access list is applied to incoming advertisements from that neighbor.
- out — that the access list is applied to outgoing advertisements to that neighbor.

Step 9 Control the number of prefixes that can be received from a neighbor:

neighbor {ipv6-address} maximum-prefix maximum [threshold][restart restart interval][warning-only]

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 maximum-prefix 7 75 restart 12
```

- maximum — the maximum number of prefixes allowed from this neighbor.
- (Optional) threshold — integer specifying at what percentage of maximum the router starts to generate a warning message. The range is from 1 to 100; the default is 75 (percent).
- (Optional) restart interval — integer value (in minutes) that specifies the time interval after which the BGP neighbor restarts.
- (Optional) warning-only — allows the router to generate a log message when the maximum number of prefixes is exceeded, instead of terminating the peering.

Step 10 Allow a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route:

neighbor {ipv6-address} default-originate [route-map map-name]

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 default-originate route-map example1
```

The argument map-name is the name of the route-map.The route map allows route 0.0.0.0 to be injected conditionally.

Step 11 Set the minimum interval between the sending of BGP routing updates:

neighbor {ipv6-address} advertisement-interval seconds

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 advertisement-interval 15
```

The argument seconds is the time (in seconds). Valid values are from 0 to 600.

Step 12 Remove private autonomous system numbers from outbound routing updates:

neighbor {ipv6-address} remove-private-as

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 remove-private-as
```

Step 13 Advertise the routes in the BGP table that matches the configured route-map:

neighbor {ipv6-address} advertise-map map-name {exist-map map-name |non-exist-map map-name}[check-all-paths]

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 advertise-map MAP1 exist-map MAP2
```

- advertise-map map name — the name of the route map that will be advertised if the conditions of the exist map or non-exist map are met.
- exist-map map name — the name of the exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not.
- non-exist-map map name — the name of the non-exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not.
- (Optional) check all paths — enables checking of all paths by the exist-map with a prefix in the BGP table.

Step 14 Sets the timers for a specific BGP peer or peer group.

neighbor {ipv6-address} timers keepalive holdtime min holdtime

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 timers 15 20 12
```

- keepalive — the frequency (in seconds) with which the ASA sends keepalive messages to its peer. The default is 60 seconds. Valid values are from 0 to 65535.
- holdtime — the interval (in seconds) after not receiving a keepalive message that the ASA declares a peer dead. The default is 180 seconds.
- min holdtime — the minimum interval (in seconds) after not receiving a keepalive message that the ASA declares a peer dead.

Step 15 Enable Message Digest 5 (MD5) authentication on a TCP connection between two BGP peers:

neighbor {ipv6-address} password string

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 password test
```

The argument string is a case-sensitive password of up to 25 characters when the service password-encryption command is enabled and up to 81 characters when the service password-encryption command is not enabled. The string can contain any alphanumeric characters, including spaces.

Note When you set the first character of the password as a number, do not provide a space immediately after the number. That is, you cannot specify a password in the format number-space-anything. The space after the number can cause authentication to fail.

Step 16 Specify that communities attributes should be sent to a BGP neighbor:

neighbor {ipv6-address} send-community [standard]

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 send-community
```

(Optional) standard keyword — only standard communities will be sent.

Step 17 Configure the router as the next hop for a BGP-speaking neighbor or peer group:

neighbor {ipv6-address}next-hop-self

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 next-hop-self
```

Step 18 Accept and attempt BGP connections to external peers residing on networks that are not directly connected:

neighbor {ipv6-address} ebgp-multihop [ttl]

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 ebgp-multihop 5
```

The argument ttl specifies time-to-live in the range from 1 to 255 hops.

Step 19 Disable connection verification to establish an eBGP peering session with a single-hop peer that uses a loopback interface:

neighbor {ipv6-address} disable-connected-check

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 disable-connected-check
```

Step 20 Secure a BGP peering session and configures the maximum number of hops that separate two external BGP (eBGP) peers:

neighbor {ipv6-address} ttl-security hops hop-count

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 ttl-security hops 15
```

The argument hop-count is the number of hops that separate the eBGP peers. The TTL value is calculated by the router from the configured hop-count argument. Valid values are from 1 to 254.

Step 21 Assign a weight to a neighbor connection:

neighbor {ipv6-address} weight number

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 weight 30
```

The argument number is the weight to assign to a neighbor connection. Valid values are from 0 to 65535.

Step 22 Configure the ASA to accept only a particular BGP version:

neighbor {ipv6-address} version number

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 version 4
```

The argument number specifies the BGP version number. The default is Version 4. Currently only BGP version 4 is supported.

Step 23 Enable a TCP transport session option for a BGP session:

neighbor {ipv6-address} transport {connection-mode{active|passive}| path-mtu-discovery[disable]}

Example:

```
ciscoasa(config-router-af)# neighbor 2000::1 transport connection-mode active
```

- connection-mode — the type of connection (active or passive).
- path-mtu-discovery — enables TCP transport path maximum transmission unit (MTU) discovery. TCP path MTU discovery is enabled by default.
- (Optional) disable — disables TCP path MTU discovery.

Step 24 Customize the AS_PATH attribute for routes received from an external Border Gateway Protocol (eBGP) neighbor:

neighbor {ipv6-address} local-as [autonomous-system-number[no-prepend]]

Example:

```
ciscoasa(config-router-af)# neighbor 10.86.118.12 local-as 5 no-prepend replace-as
```

- (Optional) autonomous-system-number — the number of an autonomous system to prepend to the AS_PATH attribute. The range of values for this argument is any valid autonomous system number from 1 to 4294967295 or 1.0 to XX.YY.
- (Optional) no-prepend — does not prepend the local autonomous system number to any routes received from the eBGP neighbor.

Caution BGP prepends the autonomous system number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This command should be configured only for autonomous system migration, and should be removed after the transition has been completed. This procedure should be attempted only by an experienced network operator. Routing loops can be created through improper configuration.

Configure IPv6 Network Settings

This section describes the steps required to define the networks to be advertised by the BGP routing process.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp *autonomous-num*

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 [unicast]

Step 3 Specify the networks to be advertised by the BGP routing processes:

network *ipv6_prefix*/*prefix_length* [**route-map** *route_map_name*]

Example:

```
ciscoasa(config-router-af)# network 2001:1/64 route-map test_route_map
```

- *ipv6 network*/*prefix_length*— The network that BGP will advertise.
- (Optional) **route-map** *name*— The identifier of a configured route map. The route map should be examined to filter the networks to be advertised. If not specified, all networks are advertised.

Configure IPv6 Redistribution Settings

This section describes the steps required to define the conditions for redistributing routes from another routing domain into BGP.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 [unicast]

Example:

```
ciscoasa(config-router)# address-family ipv6[unicast]
```

Step 3 Redistribute routes from another routing domain into a BGP autonomous system:

```
redistribute protocol  [process-id][autonomous-num][metric metric value][match{internal|
external1|external2|NSSA external 1|NSSA external 2}][route-map [map-tag]][subnets]
```

Example:

```
ciscoasa(config-router-af)# redistribute ospf 2 route-map example1 match external
```

- protocol — the source protocol from which routes are being redistributed. It can be one of the following: Connected, EIGRP, OSPF, RIP or Static.
- (Optional) process-id — For the ospf protocol, this is an appropriate OSPF process ID from which routes are to be redistributed. This identifies the routing process. This value takes the form of a nonzero decimal number.

  Note This value is auto-populated for the other protocols.

- (Optional) metric metric value — When redistributing from one OSPF process to another OSPF process on the same router, the metric will be carried through from one process to the other if no metric value is specified. When redistributing other processes to an OSPF process, the default metric is 20 when no metric value is specified. The default value is 0.
- (Optional) match internal | external1 | external2 | NSSA external 1 | NSSA external 2 — For the criteria by which OSPF routes are redistributed into other routing domains. It can be one of the following:
  - internal — Routes that are internal to a specific autonomous system.
  - external 1 — Routes that are external to the autonomous system, but are imported into BGP as OSPF Type 1 external route.
  - external 2 — Routes that are external to the autonomous system, but are imported into BGP as OSPF Type 2 external route.

- NSSA external 1 — Routes that are external to the autonomous system, but are imported into BGP as OSPF NSSA Type 1 external route.
- NSSA external 2 — Routes that are external to the autonomous system, but are imported into BGP as OSPF NSSA Type 2 external route.

- (Optional) map-tag — the identifier of a configured route map.

Note The route map should be examined to filter the networks to be redistributed. If not specified, all networks are redistributed

Configure IPv6 Route Injection Settings

This section describes the steps required to define the routes to be conditionally injected into the BGP routing table.

Procedure

Step 1 Enable a BGP routing process, which places the ASA in router configuration mode:

router bgp autonomous-num

Example:

```
ciscoasa(config)# router bgp 2
```

Step 2 Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

address-family ipv6 [unicast]

Example:

```
ciscoasa(config-router)# address-family ipv6 [unicast]
```

Step 3 Configure conditional route injection to inject more specific routes into a BGP routing table:

bgp inject-map inject-map exist-map exist-map [copy-attributes]

Example:

```
ciscoasa(config-router-af)# bgp inject-map example1 exist-map example2 copy-attributes
```

- inject-map — the name of the route map that specifies the prefixes to inject into the local BGP routing table.
- exist-map — the name of the route map containing the prefixes that the BGP speaker will track.

- (Optional) copy-attributes — configures the injected route to inherit attributes of the aggregate route.

Monitoring BGP

You can use the following commands to monitor the BGP routing process. For examples and descriptions of the command output, see the command reference. Additionally, you can disable the logging of neighbor change messages and neighbor warning messages.

To monitor various BGP routing statistics, enter one of the following commands:

- **show bgp** [ip-address [mask [longer-prefixes [injected] | shorter-prefixes [length]]]| prefix-list name | route-map name]

  Displays the entries in the BGP routing table.

- **show bgp** cidr-only

  Displays routes with non-natural network masks (that is, classless interdomain routing, or CIDR).

- **show bgp community community-number [exact-match][no-advertise][no-export]**

  Display routes that belong to specified BGP communities.

- **show bgp community-list community-list-name [exact-match]**

  Displays routes that are permitted by the BGP community list.

- **show bgp filter-list** *access-list-number*

  Displays routes that conform to a specified filter list.

- **show bgp injected-paths**

  Displays all the injected paths in the BGP routing table.

- **show bgp ipv4 unicast**

  Displays entries in the IP version 4 (IPv4) BGP routing table for unicast sessions.

- show bgp ipv6 unicast

  Displays entries in the IPv6 Border Gateway Protocol (BGP) routing table.

- show bgp ipv6 community

  Displays routes that belong to specified IPv6 Border Gateway Protocol (BGP) communities.

- show bgp ipv6 community-list

  Displays routes that are permitted by the IPv6 Border Gateway Protocol (BGP) community list.

- show bgp ipv6 filter-list

  Display routes that conform to a specified IPv6 filter list.

- show bgp ipv6 inconsistent-as

  Displays IPv6 Border Gateway Protocol (BGP) routes with inconsistent originating autonomous systems.

- show bgp ipv6 neighbors

Displays information about IPv6 Border Gateway Protocol (BGP) connections to neighbors.

- show bgp ipv6 paths

  Displays all the IPv6 Border Gateway Protocol (BGP) paths in the database.

- show bgp ipv6 prefix-list

  Displays routes that match a prefix list.

- show bgp ipv6 quote-regexp

  Displays IPv6 Border Gateway Protocol (BGP) routes matching the autonomous system path regular expression as a quoted string of characters.

- show bgp ipv6 regexp

  Displays IPv6 Border Gateway Protocol (BGP) routes matching the autonomous system path regular expression.

- show bgp ipv6 route-map

  Displays IPv6 Border Gateway Protocol (BGP) routes that failed to install in the routing table.

- show bgp ipv6 summary

  Displays the status of all IPv6 Border Gateway Protocol (BGP) connections.

- show bgp neighbors ip_address

  Displays information about BGP and TCP connections to neighbors.

- show bgp paths [LINE]

  Displays all the BGP paths in the database.

- show bgp pending-prefixes

  Displays prefixes that are pending deletion.

- show bgp prefix-list prefix_list_name [WORD]

  Displays routes that match a specified prefix list.

- show bgp regexp regexp

  Displays routes that match the autonomous system path regular expression.

- show bgp replication [index-group | ip-address]

  Displays update replication statistics for BGP update groups.

- show bgp rib-failure

  Displays BGP routes that failed to install in the Routing Information Base (RIB) table.

- show bgp route-map map-name

  Displays entries in the BGP routing table, based on the route map specified.

- show bgp summary

  Display the status of all BGP connections.

- show bgp system-config

Display the system context specific BGP configuration in multi-context mode.

This command is available in all user contexts in multi-context mode.

- show bgp update-group

  Display information about the BGP update groups.

Note To disable BGP Log messages, enter the **no bgp log-neighbor-changes** command in the router configuration mode. This disables the logging of neighbor change messages. Enter this command in router configuration mode for the BGP routing process. By default, neighbor changes are logged.

Example for BGP

This example shows how to enable and configure BGPv4 with various optional processes.

1. Define the conditions for redistributing routes from one routing protocol into another, or enable policy routing:

```
ciscoasa(config)# route-map mymap2 permit 10
```

2. Redistribute any routes that have a route address or match packet that is passed by one of the access lists specified:

```
ciscoasa(config-route-map)# match ip address acl_dmz1 acl_dmz2
```

3. Indicate where to output packets that pass a match clause of a route map for policy routing:

```
ciscoasa(config-route-map)# set ip next-hop peer address
```

4. Enable a BGP routing process from the global configuration mode:

```
ciscoasa(config)# router bgp 2
```

5. Configure a fixed router ID for the local Border Gateway Protocol (BGP) routing process in the address family configuration mode:

```
ciscoasa(config)# address-family ipv4
ciscoasa(config-router-af)# bgp router-id 19.168.254.254
```

6. Add an entry to the BGP neighbor table:

```
ciscoasa(config-router-af)# neighbor 10.108.0.0 remote-as 65
```

7. Apply a route map to incoming or outgoing routes:

```
ciscoasa(config-router-af)# neighbor 10.108.0.0 route-map mymap2 in
```

This example shows how to enable and configure BGPv6 with various optional processes.

1. Define the conditions for redistributing routes from one routing protocol into another, or enable policy routing:

   ```
   ciscoasa(config)# route-map mymap1 permit 10
   ```

2. Redistribute any routes that have a route address or match packet that is passed by one of the access lists specified:

   ```
   ciscoasa(config-route-map)# match ipv6 address acl_dmz1 acl_dmz2
   ```

3. Indicate where to output packets that pass a match clause of a route map for policy routing:

   ```
   ciscoasa(config-route-map)# set ipv6 next-hop peer address
   ```

4. Enable a BGP routing process from the global configuration mode:

   ```
   ciscoasa(config)# router bgp 2
   ```

5. Configure a fixed router ID for the local Border Gateway Protocol (BGP) routing process in the address family configuration mode:

   ```
   ciscoasa(config)# address-family ipv4
   ciscoasa(config-router-af)# bgp router-id 19.168.254.254
   ```

6. Enter address family configuration mode to configure a routing session using standard IP Version 6 (IPv6) address prefixes:

   ```
   address-family ipv6 [unicast]
   ```

7. Add an entry to the BGP neighbor table:

   ```
   ciscoasa(config-router-af)# neighbor 2001:DB8:0:CC00::1 remote-as 64600
   ```

8. Apply a route map to incoming or outgoing routes:

   ```
   ciscoasa(config-router-af)# neighbor 2001:DB8:0:CC00::1 route-map mymap1 in
   ```

History for BGP

Table 28: Feature History for BGP

Feature Name	Platform Releases	Feature Information
BGP Support	9.2(1)	

Feature Name	Platform Releases	Feature Information
		Support was added for routing data, performing authentication, and redistributing and monitoring routing information using the Border Gateway Protocol. We introduced the following commands: **router bgp, bgp maxas-limit, bgp log-neighbor-changes, bgp transport path-mtu-discovery, bgp fast-external-fallover, bgp enforce-first-as, bgp asnotation dot, timers bgp, bgp default local-preference, bgp always-compare-med, bgp bestpath compare-routerid, bgp deterministic-med, bgp bestpath med missing-as-worst, policy-list, match as-path, match community, match metric, match tag, as-path access-list, community-list, address-family ipv4, bgp router-id, distance bgp, table-map, bgp suppress-inactive, bgp redistribute-internal, bgp scan-time, bgp nexthop, aggregate-address, neighbor, bgp inject-map, show bgp, show bgp cidr-only, show bgp all community, show bgp all neighbors, show bgp community, show bgp community-list, show bgp filter-list, show bgp injected-paths, show bgp ipv4 unicast, show bgp neighbors, show bgp paths, show bgp pending-prefixes, show bgp prefix-list, show bgp regexp, show bgp replication, show bgp rib-failure, show bgp route-map, show bgp summary, show bgp system-config, show bgp update-group, clear route network, maximum-path, network.** We modified the following commands: **show route**, **show route summary**, **show running-config router**, **clear config router**, **clear route all**, **timers lsa arrival**, **timers pacing**, **timers throttle**, **redistribute bgp**.
BGP support for ASA clustering	9.3(1)	We added support for L2 and L3 clustering. We introduced the following command: bgp router-id clusterpool

Feature Name	Platform Releases	Feature Information
BGP support for nonstop forwarding	9.3(1)	We added support for Nonstop Forwarding. We introduced the following commands: bgp graceful-restart, neighbor ha-mode graceful-restart
BGP support for advertised maps	9.3(1)	We added support for BGPv4 advertised map. We introduced the following command: neighbor advertise-map
BGP support for IPv6	9.3(2)	We added support for IPv6. We introduced the following commands: address-family ipv6, ipv6 prefix-list, ipv6 prefix-list description, ipv6 prefix-list sequence-number, match ipv6 next-hop, match ipv6 route-source, match ipv6-address prefix-list, set ipv6-address prefix -list, set ipv6 next-hop, set ipv6 next-hop peer-address We modified the following command: bgp router-id

CHAPTER 27

OSPF

This chapter describes how to configure the Cisco ASA to route data, perform authentication, and redistribute routing information using the Open Shortest Path First (OSPF) routing protocol.

About OSPF

OSPF is an interior gateway routing protocol that uses link states rather than distance vectors for path selection. OSPF propagates link-state advertisements rather than routing table updates. Because only LSAs are exchanged instead of the entire routing tables, OSPF networks converge more quickly than RIP networks.

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly, rather than gradually, as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. The ASA calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

The ASA can run two processes of OSPF protocol simultaneously on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

You can redistribute routes into an OSPF routing process from another OSPF routing process, a RIP routing process, or from static and connected routes configured on OSPF-enabled interfaces.

The ASA supports the following OSPF features:

- Intra-area, inter-area, and external (Type I and Type II) routes.
- Virtual links.
- LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Configuring the ASA as a designated router or a designated backup router. The ASA also can be set up as an ABR.
- Stub areas and not-so-stubby areas.
- Area boundary router Type 3 LSA filtering.

OSPF supports MD5 and clear text neighbor authentication. Authentication should be used with all routing protocols when possible because route redistribution between OSPF and other protocols (such as RIP) can potentially be used by attackers to subvert routing information.

If NAT is used, if OSPF is operating on public and private areas, and if address filtering is required, then you need to run two OSPF processes—one process for the public areas and one for the private areas.

A router that has interfaces in multiple areas is called an Area Border Router (ABR). A router that acts as a gateway to redistribute traffic between routers using OSPF and routers using other routing protocols is called an Autonomous System Boundary Router (ASBR).

An ABR uses LSAs to send information about available routes to other OSPF routers. Using ABR Type 3 LSA filtering, you can have separate private and public areas with the ASA acting as an ABR. Type 3 LSAs (inter-area routes) can be filtered from one area to other, which allows you to use NAT and OSPF together without advertising private networks.

Note Only Type 3 LSAs can be filtered. If you configure the ASA as an ASBR in a private network, it will send Type 5 LSAs describing private networks, which will get flooded to the entire AS, including public areas.

If NAT is employed but OSPF is only running in public areas, then routes to public networks can be redistributed inside the private network, either as default or Type 5 AS external LSAs. However, you need to configure static routes for the private networks protected by the ASA. Also, you should not mix public and private networks on the same ASA interface.

You can have two OSPF routing processes, one RIP routing process, and one EIGRP routing process running on the ASA at the same time.

OSPF Support for Fast Hello Packets

The OSPF Support for Fast Hello Packets feature provides a way to configure the sending of hello packets in intervals less than one second. Such a configuration would result in faster convergence in an Open Shortest Path First (OSPF) network.

Prerequisites for OSPF Support for Fast Hello Packets

OSPF must be configured in the network already or configured at the same time as the OSPF Support for Fast Hello Packets feature.

About OSPF Support for Fast Hello Packets

The key concepts related to OSPF support for fast hello packets and the benefits of OSPF Fast Hello Packets are described below:

OSPF Hello Interval and Dead Interval

OSPF hello packets are packets that an OSPF process sends to its OSPF neighbors to maintain connectivity with those neighbors. The hello packets are sent at a configurable interval (in seconds). The defaults are 10 seconds for an Ethernet link and 30 seconds for a non broadcast link. Hello packets include a list of all neighbors for which a hello packet has been received within the dead interval. The dead interval is also a configurable interval (in seconds), and defaults to four times the value of the hello interval. The value of all hello intervals must be the same within a network. Likewise, the value of all dead intervals must be the same within a network.

These two intervals work together to maintain connectivity by indicating that the link is operational. If a router does not receive a hello packet from a neighbor within the dead interval, it will declare that neighbor to be down.

OSPF Fast Hello Packets

OSPF fast hello packets refer to hello packets being sent at intervals of less than 1 second. To understand fast hello packets, you should already understand the relationship between OSPF hello packets and the dead interval. See OSPF Hello Interval and Dead Interval, on page 679.

OSPF fast hello packets are achieved by using the ospf dead-interval command. The dead interval is set to 1 second, and the hello-multiplier value is set to the number of hello packets you want sent during that 1 second, thus providing subsecond or "fast" hello packets.

When fast hello packets are configured on the interface, the hello interval advertised in the hello packets that are sent out this interface is set to 0. The hello interval in the hello packets received over this interface is ignored.

The dead interval must be consistent on a segment, whether it is set to 1 second (for fast hello packets) or set to any other value. The hello multiplier need not be the same for the entire segment as long as at least one hello packet is sent within the dead interval.

Benefits of OSPF Fast Hello Packets

The benefit of the OSPF Fast Hello Packets feature is that your OSPF network will experience faster convergence time than it would without fast hello packets. This feature allows you to detect lost neighbors within 1 second. It is especially useful in LAN segments, where neighbor loss might not be detected by the Open System Interconnection (OSI) physical layer and data-link layer.

Implementation Differences Between OSPFv2 and OSPFv3

OSPFv3 is not backward compatible with OSPFv2. To use OSPF to route both IPv4 and IPv6 traffic, you must run both OSPFv2 and OSPFv3 at the same time. They coexist with each other, but do not interact with each other.

The additional features that OSPFv3 provides include the following:

- Protocol processing per link.
- Removal of addressing semantics.
- Addition of flooding scope.
- Support for multiple instances per link.
- Use of the IPv6 link-local address for neighbor discovery and other features.
- LSAs expressed as prefix and prefix length.
- Addition of two LSA types.
- Handling of unknown LSA types.
- Authentication support using the IPsec ESP standard for OSPFv3 routing protocol traffic, as specified by RFC-4552.

Guidelines for OSPF

Context Mode Guidelines

OSPFv2 supports single and multiple context mode.

- OSPFv2 instances cannot form adjacencies with each other across shared interfaces because, by default, inter-context exchange of multicast traffic is not supported across shared interfaces. However, you can use the static neighbor configuration under OSPFv2 process configuration under OSPFv2 process to bring up OSPFv2 neighbourship on a shared interface.
- Inter-context OSPFv2 on separate interfaces is supported.

OSPFv3 supports single mode only.

Firewall Mode Guidelines

OSPF supports routed firewall mode only. OSPF does not support transparent firewall mode.

Failover Guidelines

OSPFv2 and OSPFv3 support Stateful Failover.

IPv6 Guidelines

- OSPFv2 does not support IPv6.
- OSPFv3 supports IPv6.

- OSPFv3 uses IPv6 for authentication.
- The ASA installs OSPFv3 routes into the IPv6 RIB, provided it is the best route.
- OSPFv3 packets can be filtered out using IPv6 ACLs in the **capture** command.

OSPFv3 Hello Packets and GRE

Typically, OSPF traffic does not pass through GRE tunnel. When OSPFv3 on IPv6 is encapsulated inside GRE, the IPv6 header validation for security check such as Multicast Destination fails. The packet is dropped due to the implicit security check validation, as this packet has destination IPv6 multicast.

You may define a pre-filter rule to bypass GRE traffic. However, with pre-filter rule, inner packets would not be interrogated by the inspection engine.

Clustering Guidelines

- OSPFv3 encryption is not supported. An error message appears if you try to configure OSPFv3 encryption in a clustering environment.
- In Spanned interface mode, dynamic routing is not supported on management-only interfaces.
- In Individual interface mode, make sure that you establish the control and data units as either OSPFv2 or OSPFv3 neighbors.
- In Individual interface mode, OSPFv2 adjacencies can only be established between two contexts on a shared interface on the control unit. Configuring static neighbors is supported only on point-to point-links; therefore, only one neighbor statement is allowed on an interface.
- When a control role change occurs in the cluster, the following behavior occurs:
    - In spanned interface mode, the router process is active only on the control unit and is in a suspended state on the data units. Each cluster unit has the same router ID because the configuration has been synchronized from the control unit. As a result, a neighboring router does not notice any change in the router ID of the cluster during a role change.
    - In individual interface mode, the router process is active on all the individual cluster units. Each cluster unit chooses its own distinct router ID from the configured cluster pool. A control role change in the cluster does not change the routing topology in any way.

Multiprotocol Label Switching (MPLS) and OSPF Guidelines

When a MPLS-configured router sends Link State (LS) update packets containing opaque Type-10 link-state advertisements (LSAs) that include an MPLS header, authentication fails and the appliance silently drops the update packets, rather than acknowledging them. Eventually the peer router will terminate the neighbor relationship because it has not received any acknowledgments.

Disable the opaque capability on the ASA to ensure that the neighbor relationship remains stable:

```
router ospf process_ID_number
no nsf ietf helper
no capability opaque
```

Route Redistribution Guidelines

Redistribution of route maps with IPv4 or IPv6 prefix list on OSPFv2 or OSPFv3 is not supported. Use connected routes on OSPF for redistribution.

Additional Guidelines

- OSPFv2 and OSPFv3 support multiple instances on an interface.
- OSPFv3 supports encryption through ESP headers in a non-clustered environment.
- OSPFv3 supports Non-Payload Encryption.
- OSPFv2 supports Cisco NSF Graceful Restart and IETF NSF Graceful Restart mechanisms as defined in RFCs 4811, 4812 & 3623 respectively.
- OSPFv3 supports Graceful Restart mechanism as defined in RFC 5187.
- There is a limit to the number of intra area (type 1) routes that can be distributed. For these routes, a single type-1 LSA contains all prefixes. Because the system has a limit of 35 KB for packet size, 3000 routes result in a packet that exceeds the limit. Consider 2900 type 1 routes to be the maximum number supported.
- To avoid adjacency flaps due to route updates being dropped if the route update is larger than the minimum MTU on the link, ensure that you configure the same MTU on the interfaces on both sides of the link.

Configure OSPFv2

This section describes how to enable an OSPFv2 process on the ASA.

After you enable OSPFv2, you need to define a route map. For more information, see Define a Route Map, on page 633. Then you generate a default route. For more information, see Configure a Static Route, on page 613.

After you have defined a route map for the OSPFv2 process, you can customize it for your particular needs, To learn how to customize the OSPFv2 process on the ASA, see Customize OSPFv2, on page 685.

To enable OSPFv2, you need to create an OSPFv2 routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

You can enable up to two OSPFv2 process instances. Each OSPFv2 process has its own associated areas and networks.

To enable OSPFv2, perform the following steps:

Procedure

Step 1 Create an OSPF routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

If there is only one OSPF process enabled on the ASA, then that process is selected by default. You cannot change the OSPF process ID when editing an existing area.

Step 2 Define the IP addresses on which OSPF runs and the area ID for that interface:

network *ip_address mask* **area** *area_id*

Example:

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

When adding a new area, enter the area ID. You can specify the area ID as either a decimal number or an IP address. Valid decimal values range from 0 to 4294967295. You cannot change the area ID when editing an existing area.

Configure OSPFv2 Router ID

The OSPF Router-ID is used to identify a specific device within an OSPF database. No two routers in an OSPF system can have the same router-id.

If a router-id is not configured manually in the OSPF routing process the router will automatically configure a router-id determined from the highest IP address of an active interface. When configuring a router-id, the neighbors will not be updated automatically until that router has failed or the OSPF process has been cleared and the neighbor relationship has been re-established.

Manually Configure OSPF Router-ID

This section describes how to manually configure router-id in OSPFv2 process on the ASA.

Procedure

Step 1 To use a fixed router ID, use the **router-id** command.

router-id *ip-address*

Example:

```
ciscoasa(config-router)# router-id 193.168.3.3
```

Step 2 To revert to the previous OSPF router ID behavior, use the **no router-id** command.

no router-id *ip-address*

Example:

```
ciscoasa(config-router)# no router-id 193.168.3.3
```

Router ID Behaviour while Migrating

While migrating OSPF configuration from one ASA , say ASA 1 to another ASA, say ASA 2, the following router id selection behaviour is observed:

1. ASA 2 does not use any IP address for OSPF router-id when all interfaces are in shutdown mode. The possibilities for configuring router-id when all interfaces are in "admin down" state or shutdown mode are:

   - If ASA 2 does not have any router-id configured before, you would see this message:

     ```
     %OSPF: Router process 1 is not running, please configure a router-id
     ```

     After the first interface is brought up, ASA 2 will take IP address of this interface as router id.

   - If ASA 2 had router-id configured before and all interfaces were in "admin down" state when "no router-id" command was issued, ASA 2 will use old router id. ASA 2 uses the old router id, even if IP addresses on the interface that is brought up is changed, until "clear ospf process" command is issued.

2. ASA 2 uses new router id, when ASA 2 had router-id configured before and at least one of interfaces were not in "admin down" state or shutdown mode when "no router-id" command was issued. ASA 2 will use new router id from the IP address of the interfaces even when interfaces are in "down/down" state.

Configure OSPF Fast Hello Packets

This section describes how to configure OSPF Fast Hello Packets.

Procedure

Step 1 Configure an interface:

interface *port-channel number*

Example:

```
ciscoasa(config)# interface port-channel 10
```

The *number* argument indicates the port-channel interface number.

Step 2 Set the interval during which at least one hello packet must be received, or else the neighbor is considered down:

ospf dead-interval minimal hello-multiplier *no.of times*

Example:

```
ciscoasa(config-if)# ospf dead-interval minimal hell0-multiplier 5
ciscoasa
```

The no. of times argument indicates the number of hello packets to be sent every second. Valid values are between 3 and 20.

In this example, OSPF Support for Fast Hello Packets is enabled by specifying the minimal keyword and the hello-multiplier keyword and value. Because the multiplier is set to 5, five hello packets will be sent every second.

Customize OSPFv2

This section explains how to customize the OSPFv2 processes.

Redistribute Routes Into OSPFv2

The ASA can control the redistribution of routes between OSPFv2 routing processes.

Note If you want to redistribute a route by defining which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process, you must first generate a default route. See Configure a Static Route, on page 613, and then define a route map according to Define a Route Map, on page 633.

To redistribute static, connected, RIP, or OSPFv2 routes into an OSPFv2 process, perform the following steps:

Procedure

Step 1 Create an OSPF routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Redistribute connected routes into the OSPF routing process:

redistribute connected [[**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

Example:

```
ciscoasa(config)# redistribute connected 5 type-1 route-map-practice
```

Step 3 Redistribute static routes into the OSPF routing process:

redistribute static [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*

Example:

```
ciscoasa(config)# redistribute static 5 type-1 route-map-practice
```

Step 4 Redistribute routes from an OSPF routing process into another OSPF routing process:

redistribute ospf pid [**match {internal** | **external** [**1** | **2]** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

Example:

```
ciscoasa(config)# route-map 1-to-2 permit
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-route-map)# router ospf 2
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

You can either use the **match** options in this command to match and set route properties, or you can use a route map. The **subnets** option does not have equivalents in the **route-map** command. If you use both a route map and **match** options in the **redistribute** command, then they must match.

The example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The ASA redistributes these routes as external LSAs with a metric of 5 and a metric type of Type 1.

Step 5 Redistribute routes from a RIP routing process into the OSPF routing process:

redistribute rip [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

Example:

```
ciscoasa(config)# redistribute rip 5
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

Step 6 Redistribute routes from an EIGRP routing process into the OSPF routing process:

redistribute eigrp as-num [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

Example:

```
ciscoasa(config)# redistribute eigrp 2
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

Configure Route Summarization When Redistributing Routes Into OSPFv2

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the ASA to advertise a single route for all the redistributed routes that are included for a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

Routes that match the specified IP address mask pair can be suppressed. The tag value can be used as a match value for controlling redistribution through route maps.

Add a Route Summary Address

To configure the software advertisement on one summary route for all redistributed routes included for a network address and mask, perform the following steps:

Procedure

Step 1 Create an OSPF routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Set the summary address:

summary-address *ip_address mask* [**not-advertise**] [tag *tag*]

Example:

```
ciscoasa(config)# router ospf 1
ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0
```

In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement.

Configure Route Summarization Between OSPFv2 Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router advertises networks in one area into another area. If the network numbers in an area are assigned in a way so that they are contiguous, you can configure the area boundary router to advertise a summary route that includes all the individual networks within the area that fall into the specified range.

To define an address range for route summarization, perform the following steps:

Procedure

Step 1 Create an OSPF routing process and enters router configuration mode for this OSPF process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Set the address range:

area *area-id* range *ip-address mask* [**advertise** | **not-advertise**]

Example:

```
ciscoasa(config-rtr)# area 17 range 12.1.0.0 255.255.0.0
```

In this example, the address range is set between OSPF areas.

Configure OSPFv2 Interface Parameters

You can change some interface-specific OSPFv2 parameters, if necessary. You are not required to change any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key**. If you configure any of these parameters, be sure that the configurations for all routers on your network have compatible values.

To configure OSPFv2 interface parameters, perform the following steps:

Procedure

Step 1 Create an OSPF routing process:

router ospf*process-id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process-id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Define the IP addresses on which OSPF runs and the area ID for that interface:

network*ip-address mask***area***area-id*

Example:

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

Step 3 Enter interface configuration mode:

interface*interface-name*

Example:

```
ciscoasa(config)# interface my_interface
```

Step 4 Specify the authentication type for an interface:

ospf authentication [**message-digest** | **null**]

Example:

```
ciscoasa(config-interface)# ospf authentication message-digest
```

Step 5 Assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication:

ospf authentication-key*key*

Example:

```
ciscoasa(config-interface)# ospf authentication-key cisco
```

The *key* argument can be any continuous string of characters up to 8 bytes in length.

The password created by this command is used as a key that is inserted directly into the OSPF header when the ASA software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

Step 6 Explicitly specify the cost of sending a packet on an OSPF interface:

ospf cost*cost*

Example:

```
ciscoasa(config-interface)# ospf cost 20
```

The *cost* is an integer from 1 to 65535.

In this example, the cost is set to 20.

Step 7 Set the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet:

ospf dead-interval*seconds*

Example:

```
ciscoasa(config-interface)# ospf dead-interval 40
```

The value must be the same for all nodes on the network.

Step 8 Specify the length of time between the hello packets that the ASA sends on an OSPF interface:

ospf hello-interval*seconds*

Example:

```
ciscoasa(config-interface)# ospf hello-interval 10
```

The value must be the same for all nodes on the network.

Step 9 Enable OSPF MD5 authentication:

ospf message-digest-key*key-id***md5***key*

Example:

```
ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco
```

The following argument values can be set:

key-id—An identifier in the range from 1 to 255.

key—An alphanumeric password of up to 16 bytes.

Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value.

We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.

Step 10 Set the priority to help determine the OSPF designated router for a network:

ospf priority *number-value*

Example:

```
ciscoasa(config-interface)# ospf priority 20
```

The *number_value* argument ranges from 0 to 255.

In multiple context mode, for shared interfaces, specify 0 to ensure the device does not become the designated router. OSPFv2 instances cannot form adjacencies with each other across shared interfaces.

Step 11 Specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface:

ospf retransmit-interval*number-value*

Example:

```
ciscoasa(config-interface)# ospf retransmit-interval seconds
```

The value for *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 8192 seconds. The default value is 5 seconds.

Step 12 Set the estimated number of seconds required to send a link-state update packet on an OSPF interface:

ospf transmit-delay*seconds*

Example:

```
ciscoasa(config-interface)# ospf transmit-delay 5
```

The *seconds* value ranges from 1 to 8192 seconds.The default value is 1 second.

Step 13 Set the number of hello packets sent during 1 second:

ospf dead-interval minimal hello-interval multiplier*integer*

Example:

```
ciscoasa(config-if)# ospf dead-interval minimal hello-multiplier 6
```

Valid values are integers between 3 and 20.

Step 14 Specify the interface as a point-to-point, non-broadcast network:

ospf network point-to-point non-broadcast

Example:

```
ciscoasa(config-interface)# ospf network point-to-point non-broadcast
```

When you designate an interface as point-to-point and non-broadcast, you must manually define the OSPF neighbor; dynamic neighbor discovery is not possible. See Define Static OSPFv2 Neighbors, on page 695 for more information. Additionally, you can only define one OSPF neighbor on that interface.

Configure OSPFv2 Area Parameters

You can configure several OSPF area parameters. These area parameters (shown in the following task list) include setting authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in the stub area. To further reduce the number of LSAs sent into a stub area, you can use the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending a summary link advertisement (LSA Type 3) into the stub area.

Procedure

Step 1 Create an OSPF routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Enable authentication for an OSPF area:

area *area-id* authentication

Example:

```
ciscoasa(config-rtr)# area 0 authentication
```

Step 3 Enable MD5 authentication for an OSPF area:

area *area-id* authentication message-digest

Example:

```
ciscoasa(config-rtr)# area 0 authentication message-digest
```

Configure OSPFv2 Filter Rules

Use the following procedure to filter routes or networks received or transmitted in OSPF updates.

Procedure

Step 1 Enable an OSPF routing process and enter router configuration mode:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

Step 2 Filter routes or networks received in incoming or advertised in outgoing OSPF updates:

distribute-list *acl-number* **in** [**interface** *ifname*]

distribute-list *acl-number* **out** [*protocol process-number* | **connected** | **static**]

The argument *acl-number* specifies IP access list number. The access list defines which networks are to be received and which are to be suppressed in routing updates.

To apply the filter to incoming updates, specify **in**. You can optionally specify an interface to limit the filter to updates received on that interface.

To apply the filter to outbound updates, specify **out**. You can optionally specify a protocol (**bgp**, **eigrp**, **ospf**, or **rip**) with a process number (except for RIP) to apply to the distribution list. You can also filter on whether the peers and networks were learned through **connected** or **static** routes.

Example:

```
ciscoasa(config-rtr)# distribute-list ExampleAcl in interface inside
```

Configure an OSPFv2 NSSA

The OSPFv2 implementation of an NSSA is similar to an OSPFv2 stub area. NSSA does not flood Type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

NSSA imports Type 7 autonomous system external routes within an NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA ABRs, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

You can simplify administration if you are an ISP or a network administrator that must connect a central site using OSPFv2 to a remote site that is using a different routing protocol with NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPFv2 stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPFv2 to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

Before you use this feature, consider these guidelines:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the router generates a Type 7 default into the NSSA or the NSSA area boundary router.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers cannot communicate with each other.

Procedure

Step 1 Create an OSPF routing processs:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Define an NSSA area:

area *area-id* nssa [**no-redistribution**] [**default-information-originate**]

Example:

```
ciscoasa(config-rtr)# area 0 nssa
```

Step 3 Set the summary address and helps reduce the size of the routing table:

summary-address *ip_address mask* [**not-advertise**] [tag *tag*]

Example:

```
ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0
```

Using this command for OSPF causes an OSPF ASBR to advertise one external route as an aggregate for all redistributed routes that are covered by the address.

In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement.

Note OSPF does not support summary-address 0.0.0.0 0.0.0.0.

Configure an IP Address Pool for Clustering (OSPFv2 and OSPFv3)

You can assign a range of IPv4 addresses for the router ID cluster pool if you are using Individual Interface clustering.

To assign a range of IPv4 addresses for the router ID cluster pool in Individual Interface clustering for OSPFv2 and OSPFv3, enter the following command:

Procedure

Specify the router ID cluster pool for Individual Interface clustering:

router-id cluster-pool hostname | A.B.C.D ip_pool

Example:

```
hostname(config)# ip local pool rpool 1.1.1.1-1.1.1.4
hostname(config)# router ospf 1
hostname(config-rtr)# router-id cluster-pool rpool
hostname(config-rtr)# network 17.5.0.0 255.255.0.0 area 1
hostname(config-rtr)# log-adj-changes
```

The **cluster-pool** keyword enables configuration of an IP address pool when Individual Interface clustering is configured. The **hostname** | **A.B.C.D.** keyword specifies the OSPF router ID for this OSPF process. The *ip_pool* argument specifies the name of the IP address pool.

Note If you are using clustering, then you do not need to specify an IP address pool for the router ID. If you do not configure an IP address pool, then the ASA uses the automatically generated router ID.

Define Static OSPFv2 Neighbors

You need to define static OSPFv2 neighbors to advertise OSPFv2 routes over a point-to-point, non-broadcast network. This feature lets you broadcast OSPFv2 advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

Before you begin, you must create a static route to the OSPFv2 neighbor. See Configure a Static Route, on page 613 for more information about creating static routes.

Procedure

Step 1 Create an OSPFv2 routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Define the OSPFv2 neighborhood:

neighbor *addr* [**interface** if_name]

Example:

```
ciscoasa(config-rtr)# neighbor 255.255.0.0 [interface my_interface]
```

The *addr* argument is the IP address of the OSPFv2 neighbor. The *if_name* argument is the interface used to communicate with the neighbor. If the OSPF v2neighbor is not on the same network as any of the directly connected interfaces, you must specify the interface.

Configure Route Calculation Timers

You can configure the delay time between when OSPFv2 receives a topology change and when it starts an SPF calculation. You also can configure the hold time between two consecutive SPF calculations.

Procedure

Step 1 Create an OSPFv2 routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Configure the route calculation times:

timers throttle spf *spf-start spf-hold spf-maximum*

Example:

```
ciscoasa(config-router)# timers throttle spf 500 500 600
```

The *spf-start* argument is the delay time (in milliseconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 600000.

The *spf-hold* argument is the minimum time (in milliseconds) between two consecutive SPF calculations. It can be an integer from 0 to 600000.

The spf-maximum argument is the maximum time (in milliseconds) between two consecutive SPF calculations. It can be integer from 0 to 600000.

Log Neighbors Going Up or Down

By default, a syslog message is generated when an OSPFv2 neighbor goes up or down.

Configure the **log-adj-changes** command if you want to know about OSPFv2 neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** command provides a higher level view of the peer relationship with less output. Configure the **log-adj-changes detail** command if you want to see messages for each state change.

Procedure

Step 1 Create an OSPFv2 routing process:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The *process_id* argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Configure logging for neighbors going up or down:

log-adj-changes [**detail**]

Configure OSPFv3

This section describes the tasks involved in configuring an OSPFv3 routing process.

Enable OSPFv3

To enable OSPFv3, you need to create an OSPFv3 routing process, create an area for OSPFv3, enable an interface for OSPFv3, then redistribute the route into the targeted OSPFv3 routing processes.

Procedure

Step 1 Create an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config)# ipv6 router ospf 10
```

The *process-id* argument is an internally used tag for this routing process and can be any positive integer. This tag does not have to match the tag on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Enable an interface:

interface *interface_name*

Example:

```
ciscoasa(config)# interface Gigabitethernet0/0
```

Step 3 Create the OSPFv3 routing process with the specified process ID and an area for OSPFv3 with the specified area ID:

ipv6 ospf *process-id* **area** *area_id*

Example:

```
ciscoasa(config)# ipv6 ospf 200 area 100
```

Configure OSPFv3 Interface Parameters

You can change certain interface-specific OSPFv3 parameters, if necessary. You are not required to change any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: the hello interval and the dead interval. If you configure any of these parameters, be sure that the configurations for all routers on your network have compatible values.

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 10
```

The *process-id* argument is an internally used tag for this routing process and can be any positive integer. This tag does not have to match the tag on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Create an OSPFv3 area:.

ipv6 ospf **area** [*area-num*] [**instance**]

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
```

The *area-num* argument is the area for which authentication is to be enabled and can be either a decimal value or an IP address. The **instance** keyword specifies the area instance ID that is to be assigned to an interface. An interface can have only one OSPFv3 area. You can use the same area on multiple interfaces, and each interface can use a different area instance ID.

Step 3 Specify the cost of sending a packet on an interface:

ipv6 ospf cost *interface-cost*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
```

The *interface-cost* argument specifies an unsigned integer value expressed as the link-state metric, which can range in value from 1 to 65535. The default cost is based on the bandwidth.

Step 4 Filter outgoing LSAs to an OSPFv3 interface:

ipv6 ospf database-filter all out

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf database-filter all out
```

All outgoing LSAs are flooded to the interface by default.

Step 5 Set the time period in seconds for which hello packets must not be seen before neighbors indicate that the router is down:

ipv6 ospf dead-interval *seconds*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf dead-interval 60
```

The value must be the same for all nodes on the network and can range from 1 to 65535. The default is four times the interval set by the **ipv6 ospf hello-interval** command.

Step 6 Specify the encryption type for an interface:

ipv6 ospf encryption {**ipsec spi** *spi* **esp** *encryption-algorithm* [[*key-encryption-type*] *key*] *authentication-algorithm* [[*key-encryption-type*] *key* | **null**}

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf encryption ipsec spi 1001 esp null sha1 123456789A123456789B123456789C123456789D
```

The **ipsec** keyword specifies the IP security protocol. The **spi** *spi* keyword-argument pair specifies the security policy index, which must be in the range of 256 to 42949667295 and entered as a decimal.

The **esp** keyword specifies the encapsulating security payload. The *encryption-algorithm* argument specifies the encryption algorithm to be used with ESP. Valid values include the following:

- aes-cdc—Enables AES-CDC encryption.
- 3des—Enables 3DES encryption.
- des—Enables DES encryption.
- null—Specifies ESP with no encryption.

The *key-encryption-type* argument can be one of the following two values:

- 0—The key is not encrypted.
- 7—The key is encrypted.

The *key* argument specifies the number used in the calculation of the message digest. The number is 32 hexadecimal digits (16 bytes) long. The size of the key depends on the encryption algorithm used. Some algorithms, such as AES-CDC, allow you to choose the size of the key. The *authentication-algorithm* argument specifies the encryption authentication algorithm to be used, which can be one of the following:

- md5—Enables message digest 5 (MD5).
- sha1—Enables SHA-1.

The **null** keyword overrides area encryption.

If OSPFv3 encryption is enabled on an interface and a neighbor is on different area (for example, area 0), and you want the ASA to form adjacencies with that area, you must change the area on the ASA. After you have changed the area on the ASA to 0, there is a delay of two minutes before the OSPFv3 adjacency comes up.

Step 7 Specify the flood reduction of LSAs to the interface:

ipv6 ospf flood-reduction

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf flood reduction
```

Step 8 Specify the interval in seconds between hello packets sent on the interface:

ipv6 ospf hello-interval *seconds*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf hello-interval 15
```

The value must be the same for all nodes on a specific network and can range from 1 to 65535. The default interval is 10 seconds for Ethernet interfaces and 30 seconds for non-broadcast interfaces.

Step 9 Disable the OSPF MTU mismatch detection when DBD packets are received:

ipv6 ospf mtu-ignore

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf mtu-ignore
```

OSPF MTU mismatch detection is enabled by default.

Step 10 Set the OSPF network type to a type other than the default, which depends on the network type:

ipv6 ospf network {**broadcast** | **point-to-point non-broadcast**}

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf network point-to-point non-broadcast
```

The **point-to-point non-broadcast** keyword sets the network type to point-to-point non-broadcast. The **broadcast** keyword sets the network type to broadcast.

Step 11 Set the router priority, which helps determine the designated router for a network:

ipv6 ospf priority *number-value*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf priority 4
```

Valid values range from 0 to 255.

Step 12 Configure OSPFv3 router interconnections to non-broadcast networks:

ipv6 ospf neighbor *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**]

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01
```

Step 13 Specify the time in seconds between LSA retransmissions for adjacencies that belong to the interface:

ipv6 ospf retransmit-interval *seconds*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-interval 8
```

The time must be greater than the expected round-trip delay between any two routers on the attached network. Valid values range from 1 to 65535 seconds. The default is 5 seconds.

Step 14 Set the estimated time in seconds to send a link-state update packet on the interface:

ipv6 ospf transmit-delay *seconds*

Example:

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-delay 3
```

Valid values range from 1 to 65535 seconds. The default is 1 second.

Configure OSPFv3 Router Parameters

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config)# ipv6 router ospf 10
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Configure OSPFv3 area parameters:

area

Example:

```
ciscoasa(config-rtr)# area 10
```

Supported parameters include the area ID as a decimal value from 0 to 4294967295 and the area ID in the IP address format of **A.B.C.D**.

Step 3 Set a command to its default value:

default

Example:

```
ciscoasa(config-rtr)# default originate
```

The **originate** parameter distributes the default route.

Step 4 Control distribution of default information:

default-information

Step 5 Define the OSPFv3 route administrative distance based on the route type:

distance

Example:

```
ciscoasa(config-rtr)# distance 200
```

Supported parameters include the administrative distance with values from 1 to 254 and **ospf** for the OSPFv3 distance.

Step 6 Suppress the sending of syslog messages with the **lsa** parameter when the router receives a link-state advertisement (LSA) for Type 6 Multicast OSPF (MOSPF) packets:

ignore

Example:

```
ciscoasa(config-rtr)# ignore lsa
```

Step 7 Configure the router to send a syslog message when an OSPFv3 neighbor goes up or down:

log-adjacency-changes

Example:

```
ciscoasa(config-rtr)# log-adjacency-changes detail
```

With the **detail** parameter, all state changes are logged.

Step 8 Suppress the sending and receiving of routing updates on an interface:

passive-interface [*interface_name*]

Example:

```
ciscoasa(config-rtr)# passive-interface inside
```

The *interface_name* argument specifies the name of the interface on which the OSPFv3 process is running.

Step 9 Configure the redistribution of routes from one routing domain into another:

redistribute {**connected** | **ospf** | **static**}

Where:

- **connected**—Specifies connected routes.
- **ospf**—Specifies OSPFv3 routes.
- **static**—Specifies static routes.

Example:

```
 ciscoasa(config-rtr)# redistribute ospf
```

Step 10 Create a fixed router ID for a specified process:

router-id {*A.B.C.D* | **cluster-pool** | **static**}

Where:

A.B.C.D—Specifies the OSPF router ID in IP address format.

cluster-pool—Configures an IP address pool when Individual Interface clustering is configured. For more information about IP address pools used in clustering, see Configure an IP Address Pool for Clustering (OSPFv2 and OSPFv3), on page 694.

Example:

```
ciscoasa(config-rtr)# router-id 10.1.1.1
```

Step 11 Configure IPv6 address summaries with valid values from 0 to 128:

summary-prefix *X:X:X:X::X/*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
ciscoasa(config-router)# router-id 192.168.3.3
ciscoasa(config-router)# summary-prefix FECO::/24
ciscoasa(config-router)# redistribute static
```

The *X:X:X:X::X/* parameter specifies the IPv6 prefix.

Step 12 Adjust routing timers:

timers

The routing timer parameters are the following:

- **lsa**—Specifies OSPFv3 LSA timers.
- **nsf**—Specifies OSPFv3 NSF wait timers.
- **pacing**—Specifies OSPFv3 pacing timers.
- **throttle**—Specifies OSPFv3 throttle timers.

Example:

```
ciscoasa(config)# ipv6 router ospf 10
ciscoasa(config-rtr)# timers throttle spf 6000 12000 14000
```

Configure OSPFv3 Area Parameters

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535.

This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Set the summary default cost of an NSSA area or a stub area:

area *area-id* **default-cost** *cost*

Example:

```
ciscoasa(config-rtr)# area 1 default-cost nssa
```

Step 3 Summarize routes that match the address and mask for border routers only:

area *area-id* **range** *ipv6-prefix*/ *prefix-length* [**advertise** | **not advertise**] [**cost** *cost*]

Example:

```
ciscoasa(config-rtr)# area 1 range FE01:1::1/64
```

- The *area-id* argument identifies the area for which routes are to be summarized. The value can be specified as a decimal or an IPv6 prefix.
- The *ipv6-prefix* argument specifies the IPv6 prefix. The *prefix-length* argument specifies the prefix length.
- The **advertise** keyword sets the address range status to advertised and generates a Type 3 summary LSA.
- The **not-advertise** keyword sets the address range status to DoNotAdvertise.
- The Type 3 summary LSA is suppressed, and the component networks remain hidden from other networks.
- The **cost** *cost* keyword-argument pair specifies the metric or cost for the summary route, which is used during OSPF SPF calculations to determine the shortest paths to the destination.
- Valid values range from 0 to 16777215.

Step 4 Specify an NSSA area:

area *area-id* **nssa**

Example:

```
ciscoasa(config-rtr)# area 1 nssa
```

Step 5 Specify a stub area:

area *area-id* **stub**

Example:

```
ciscoasa(config-rtr)# area 1 stub
```

Step 6 Define a virtual link and its parameters:

area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**ttl-security hops** *hop-count*]

Example:

```
ciscoasa(config-rtr)# area 1 virtual-link 192.168.255.1 hello-interval 5
```

- The *area-id* argument identifies the area for which routes are to be summarized. The **virtual link** keyword specifies the creation of a virtual link neighbor.
- The *router-id* argument specifies the router ID that is associated with the virtual link neighbor.
- Enter the **show ospf** or **show ipv6 ospf** command to display the router ID. There is no default value.
- The **hello-interval** keyword specifies the time in seconds between the hello packets that are sent on an interface. The hello interval is an unsigned integer that is to be advertised in the hello packets. The value must be the same for all routers and access servers that are attached to a common network. Valid values range from 1 to 8192. The default is 10.

- The **retransmit-interval** *seconds* keyword-argument pair specifies the time in seconds between LSA retransmissions for adjacencies that belong to the interface. The retransmit interval is the expected round-trip delay between any two routers on the attached network. The value must be greater than the expected round-trip delay, and can range from 1 to 8192. The default is 5.
- The **transmit-delay** *seconds* keyword-argument pair specifies the estimated time in seconds that is required to send a link-state update packet on the interface. The integer value must be greater than zero. LSAs in the update packet have their own ages incremented by this amount before transmission. The range of values can be from 1 to 8192. The default is 1.
- The **dead-interval** *seconds* keyword-argument pair specifies the time in seconds that hello packets are not seen before a neighbor indicates that the router is down. The dead interval is an unsigned integer. The default is four times the hello interval, or 40 seconds. The value must be the same for all routers and access servers that are attached to a common network. Valid values range from 1 to 8192.
- The **ttl-security hops** keyword configures the time-to-live (TTL) security on a virtual link. The *hop-count* argument value can range from 1 to 254.

Configure OSPFv3 Passive Interfaces

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process_id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Suppress the sending and receiving of routing updates on an interface:

passive-interface [*interface_name*]

Example:

```
ciscoasa(config-rtr)# passive-interface inside
```

The *interface_name* argument specifies the name of the interface on which the OSPFv3 process is running. If the *no interface_name* argument is specified, all of the interfaces in the OSPFv3 process *process_id* are made passive.

Configure OSPFv3 Administrative Distance

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process_id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Set the administrative distance for OSPFv3 routes:

distance [**ospf** {**external** | **inter-area** | **intra-area**}] *distance*

Example:

```
ciscoasa(config-rtr)# distance ospf external 200
```

The **ospf** keyword specifies OSPFv3 routes. The **external** keyword specifies the external Type 5 and Type 7 routes for OSPFv3. The **inter-area** keyword specifies the inter-area routes for OSPVv3. The **intra-area** keyword specifies the intra-area routes for OSPFv3. The *distance* argument specifies the administrative distance, which is an integer from 10 to 254.

Configure OSPFv3 Timers

You can set LSA arrival, LSA pacing, and throttling timers for OSPFv3.

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Set the minimum interval at which the ASA accepts the same LSA from OSPF neighbors:

timers lsa arrival *milliseconds*

Example:

```
ciscoasa(config-rtr)# timers lsa arrival 2000
```

The *milliseconds* argument specifies the minimum delay in milliseconds that must pass between acceptance of the same LSA arriving from neighbors. The range is from 0 to 6,000,000 milliseconds. The default is 1000 milliseconds.

Step 3 Configure LSA flood packet pacing:

timers pacing flood *milliseconds*

Example:

```
ciscoasa(config-rtr)# timers lsa flood 20
```

The *milliseconds* argument specifies the time in milliseconds at which LSAs in the flooding queue are paced in between updates. The configurable range is from 5 to 100 milliseconds. The default value is 33 milliseconds.

Step 4 Change the interval at which OSPFv3 LSAs are collected into a group and refreshed, checksummed, or aged:

timers pacing lsa-group *seconds*

Example:

```
ciscoasa(config-rtr)# timers pacing lsa-group 300
```

The *seconds* argument specifies the number of seconds in the interval at which LSAs are grouped, refreshed, check summed, or aged. The range is from 10 to 1800 seconds. The default value is 240 seconds.

Step 5 Configure LSA retransmission packet pacing:

timers pacing retransmission *milliseconds*

Example:

```
ciscoasa(config-rtr)# timers pacing retransmission 100
```

The *milliseconds* argument specifies the time in milliseconds at which LSAs in the retransmission queue are paced. The configurable range is from 5 to 200 milliseconds. The default value is 66 milliseconds.

Step 6 Configure OSPFv3 LSA throttling:

timers throttle lsa *milliseconds1 milliseconds2 milliseconds3*

Example:

```
ciscoasa(config-rtr)# timers throttle lsa 500 6000 8000
```

- The *milliseconds1* argument specifies the delay in milliseconds to generate the first occurrence of the LSA. The *milliseconds2* argument specifies the maximum delay in milliseconds to originate the same LSA. The *milliseconds3* argument specifies the minimum delay in milliseconds to originate the same LSA.

- For LSA throttling, if the minimum or maximum time is less than the first occurrence value, then OSPFv3 automatically corrects to the first occurrence value. Similarly, if the maximum delay specified is less than the minimum delay, then OSPFv3 automatically corrects to the minimum delay value.
- For *milliseconds1*, the default value is 0 milliseconds.
- For *milliseconds2* and *milliseconds3*, the default value is 5000 milliseconds.

Step 7 Configure OSPFv3 SPF throttling:

timers throttle spf *milliseconds1 milliseconds2 milliseconds3*

Example:

```
ciscoasa(config-rtr)# timers throttle spf 5000 12000 16000
```

- The *milliseconds1* argument specifies the delay in milliseconds to receive a change to the SPF calculation. The *milliseconds2* argument specifies the delay in milliseconds between the first and second SPF calculations. The *milliseconds3* argument specifies the maximum wait time in milliseconds for SPF calculations.
- For SPF throttling, if *milliseconds2* or *milliseconds3* is less than *milliseconds1*, then OSPFv3 automatically corrects to the *milliseconds1* value. Similarly, if *milliseconds3* is less than *milliseconds2*, then OSPFv3 automatically corrects to the *milliseconds2* value.
- For *milliseconds1*, the default value of SPF throttling is 5000 milliseconds.
- For *milliseconds2* and *milliseconds3*, the default value of SPF throttling is 10000 milliseconds.

Define Static OSPFv3 Neighbors

You need to define static OSPFv3 neighbors to advertise OSPF routes over a point-to-point, non-broadcast network. This feature lets you broadcast OSPFv3 advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

Before you begin, you must create a static route to the OSPFv3 neighbor. See Configure a Static Route, on page 613 for more information about creating static routes.

Procedure

Step 1 Enable an OSPFv3 routing process and enters IPv6 router configuration mode.

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Configure OSPFv3 router interconnections to non-broadcast networks.

ipv6 ospf neighbor *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**]

Example:

```
ciscoasa(config-if)# interface ethernet0/0 ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01
```

Reset OSPFv3 Default Parameters

To return an OSPFv3 parameter to its default value, perform the following steps:

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Return an optional parameter to its default value:

default [**area** | **auto-cost** | **default-information** | **default-metric** | **discard-route** | **discard-route | distance** | **distribute-list** | **ignore** | **log-adjacency-changes** | **maximum-paths** | **passive-interface** | **redistribute** | **router-id** | **summary-prefix** | **timers**]

Example:

```
ciscoasa(config-rtr)# default metric 5
```

- The **area** keyword specifies the OSPFv3 area parameters. The **auto-cost** keyword specifies the OSPFv3 interface cost according to bandwidth.
- The **default-information** keyword distributes default information. The **default-metric** keyword specifies the metric for a redistributed route
- The **discard-route** keyword enables or disables the discard-route installation. The **distance** keyword specifies the administrative distance.
- The **distribute-list** keyword filters networks in routing updates.
- The **ignore** keyword ignores a specific event. The **log-adjacency-changes** keyword logs changes in the adjacency state.

- The **maximum-paths** keyword forwards packets over multiple paths.
- The **passive-interface** keyword suppresses routing updates on an interface.
- The **redistribute** keyword redistributes IPv6 prefixes from another routing protocol.
- The **router-id** keyword specifies the router ID for the specified routing process.
- The **summary-prefix** keyword specifies the IPv6 summary prefix.
- The **timers** keyword specifies the OSPFv3 timers.

Send Syslog Messages

Configure the router to send a syslog message when an OSPFv3 neighbor goes up or down.

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Configure the router to send a syslog message when an OSPFv3 neighbor goes up or down:

log-adjacency-changes [**detail**]

Example:

```
ciscoasa(config-rtr)# log-adjacency-changes detail
```

The **detail** keyword sends a syslog message for each state, not only when an OSPFv3 neighbor goes up or down.

Suppress Syslog Messages

To suppress the sending of syslog messages when the route receives unsupported LSA Type 6 multicast OSPF (MOSPF) packets, perform the following steps:

Procedure

Step 1 Enable an OSPFv2 routing process:

router ospf *process_id*

Example:

```
ciscoasa(config-if)# router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Suppress the sending of syslog messages when the router receives unsupported LSA Type 6 MOSPF packets:

ignore lsa mospf

Example:

```
ciscoasa(config-rtr)# ignore lsa mospf
```

Calculate Summary Route Costs

Procedure

Restore the methods that are used to calculate summary route costs according to RFC 1583:

compatible rfc1583

Example:

```
ciscoasa (config-rtr)# compatible rfc1583
```

Generate a Default External Route into an OSPFv3 Routing Domain

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Generate a default external route into an OSPFv3 routing domain:

default-information originate [**always**] **metric** *metric-value* [**metric-type** *type-value*] [**route-map** *map-name*]

Example:

```
ciscoasa(config-rtr)# default-information originate always metric 3 metric-type 2
```

- The **always** keyword advertises the default route whether or not the default route exists.
- The **metric** *metric-value* keyword-argument pair specifies the metric used for generating the default route.
- If you do not specify a value using the **default-metric** command, the default value is 10. Valid metric values range from 0 to 16777214.
- The **metric-type** *type-value* keyword-argument pair specifies the external link type that is associated with the default route that is advertised into the OSPFv3 routing domain. Valid values can be one of the following:
  - 1—Type 1 external route
  - 2—Type 2 external route

  The default is the type 2 external route.
- The **route-map** *map-name* keyword-argument pair specifies the routing process that generates the default route if the route map is satisfied.

Configure an IPv6 Summary Prefix

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Configure an IPv6 summary prefix:

summary-prefix *prefix* [**not-advertise** | **tag** *tag-value*]

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
ciscoasa(config-rtr)# router-id 192.168.3.3
ciscoasa(config-rtr)# summary-prefix FECO::/24
ciscoasa(config-rtr)# redistribute static
```

The *prefix* argument is the IPv6 route prefix for the destination. The **not-advertise** keyword suppresses routes that match the specified prefix and mask pair. This keyword applies to OSPFv3 only. The **tag** *tag-value* keyword-argument pair specifies the tag value that can be used as a match value for controlling redistribution through route maps. This keyword applies to OSPFv3 only.

Redistribute IPv6 Routes

Procedure

Step 1 Enable an OSPFv3 routing process:

ipv6 router ospf *process-id*

Example:

```
ciscoasa(config-if)# ipv6 router ospf 1
```

The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2 Redistribute IPv6 routes from one OSPFv3 process into another:

redistribute *source-protocol* [*process-id*] [**include-connected** {[**level-1** | **level-2**}] [*as-number*] [**metric** [*metric-value* | **transparent**}] **[metric-type** *type-value*] [**match** {**external** [**1**|**2**] | **internal** | **nssa-external** [**1**|**2**]}] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
ciscoasa(config-rtr)# redistribute connected 5 type-1
```

- The *source-protocol* argument specifies the source protocol from which routes are being redistributed, which can be static, connected, or OSPFv3.
- The *process-id* argument is the number that is assigned administratively when the OSPFv3 routing process is enabled.
- The **include-connected** keyword allows the target protocol to redistribute routes learned by the source protocol and connected prefixes on those interfaces over which the source protocol is running.

- The **level-1** keyword specifies that for Intermediate System-to-Intermediate System (IS-IS), Level 1 routes are redistributed into other IP routing protocols independently.
- The **level-1-2** keyword specifies that for IS-IS, both Level 1 and Level 2 routes are redistributed into other IP routing protocols.
- The **level-2** keyword specifies that for IS-IS, Level 2 routes are redistributed into other IP routing protocols independently.
- For the **metric** *metric-value* keyword-argument pair, when redistributing routes from one OSPFv3 process into another OSPFv3 process on the same router, the metric is carried through from one process to the other if no metric value is specified. When redistributing other processes into an OSPFv3 process, the default metric is 20 when no metric value is specified.
- The **metric transparent** keyword causes RIP to use the routing table metric for redistributed routes as the RIP metric.
- The **metric-type** *type-value* keyword-argument pair specifies the external link type that is associated with the default route that is advertised into the OSPFv3 routing domain. Valid values can be one of the following: 1 for a Type 1 external route or 2 for a Type 2 external route. If no value is specified for the **metric-type** keyword, the ASA adopts a Type 2 external route. For IS-IS, the link type can be one of the following: internal for an IS-IS metric that is less than 63 or external for an IS-IS metric that is greater than 64 and less than 128. The default is internal.
- The **match** keyword redistributes routes into other routing domains and is used with one of the following options: **external** [**1**|**2**] for routes that are external to the autonomous system, but are imported into OSPFv3 as Type 1 or Type 2 external routes; **internal** for routes that are internal to a specific autonomous system; **nssa-external** [**1**|**2**] for routes that are external to the autonomous system, but are imported into OSPFv3 in an NSSA for IPv6 as Type 1 or Type 2 external routes.
- The **tag** *tag-value* keyword-argument pair specifies the 32-bit decimal value that is attached to each external route, which may be used to communicate information between ASBRs. If none is specified, then the remote autonomous system number is used for routes from BGP and EGP. For other protocols, zero is used. Valid values range from 0 to 4294967295.
- The **route-map** keyword specifies the route map to check for filtering the importing of routes from the source routing protocol to the current routing protocol. If this keyword is not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes are imported. The *map-tag* argument identifies a configured route map.

Configure Graceful Restart

The ASA may experience some known failure situations, that should not affect packet forwarding across the switching platform. The Non-Stop Forwarding (NSF) capability allows data forwarding to continue along known routes, while the routing protocol information is being restored. This capability is useful when there is a component failure (i.e., active unit crash with standby unit taking over in failover (HA) mode, control unit crash with data unit elected as new control unit in cluster mode), or when there is a scheduled hitless software upgrade.

Graceful restart is supported on both OSPFv2 and OSPFv3. You can configure graceful restart on OSPFv2 by using either using NSF Cisco (RFC 4811 and RFC 4812) or NSF IETF (RFC 3623). You can configure graceful restart on OSPFv3 using graceful-restart (RFC 5187).

Configuring the NSF graceful-restart feature involves two steps; configuring capabilities and configuring a device as NSF-capable or NSF-aware. A NSF-capable device can indicate its own restart activities to neighbors and a NSF-aware device can help a restarting neighbor.

A device can be configured as NSF-capable or NSF-aware, depending on some conditions:

- A device can be configured as NSF-aware irrespective of the mode in which it is.
- A device has to be in either Failover or Spanned Etherchannel (L2) cluster mode to be configured as NSF-capable.
- For a device to be either NSF-aware or NSF-capable, it should be configured with the capability of handling opaque Link State Advertisements (LSAs)/ Link Local Signaling (LLS) block as required.

Note When fast hellos are configured for OSPFv2, graceful restart does not occur when the active unit reloads and the standby unit becomes active. This is because the time taken for the role change is more than the configured dead interval.

Configure Capabilities

The Cisco NSF Graceful Restart mechanism depends on the LLS capability as it sends an LLS block with the RS-bit set in the Hello packet, to indicate the restart activity. The IETF NSF mechanism depends on the opaque LSA capability as it sends opaque-LSAs of type-9 to indicate the restart activity. To configure capabilities enter the following commands:

Procedure

Step 1 Create an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute:

router ospf *process_id*

Example:

```
ciscoasa(config)# router ospf 2
```

The process_id argument is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 Enable the use of LLS data block or opaque LSAs to enable NSF:

capability {lls|opaque}

The lls keyword is used to enable LLS capability for Cisco NSF Graceful Restart mechanism.

The opaque keyword is used to enable opaque LSA capability for IETF NSF Graceful Restart mechanism.

Configuring Graceful Restart for OSPFv2

There are two graceful restart mechanisms for OSPFv2, Cisco NSF and IETF NSF. Only one of these graceful restart mechanisms can be configured at a time for an ospf instance. An NSF-aware device can be configured as both Cisco NSF helper and IETF NSF helper but a NSF-capable device can be configured in either Cisco NSF or IETF NSF mode at a time for an ospf instance.

Configure Cisco NSF Graceful Restart for OSPFv2

Configure Cisco NSF Graceful Restart for OSPFv2, for a NSF-capable or NSF-aware device.

Procedure

Step 1 Enable Cisco NSF on a NSF-capable device:

nsf cisco [enforce global]

Example:

```
ciscoasa(config-router)# nsf cisco
```

The enforce global keyword cancels NSF restart when non-NSF-aware neighbor devices are detected.

Step 2 Enable Cisco NSF helper mode on NSF-aware device:

capability {lls|opaque}

Example:

```
ciscoasa(config-router)# capability lls
```

This command is enabled by default. Using the no form of the command disables it.

Configure IETF NSF Graceful Restart for OSPFv2

Configure IETF NSF Graceful Restart for OSPFv2, for a NSF-capable or NSF-aware device.

Procedure

Step 1 Enable IETF NSF on a NSF-capable device:

nsf ietf [restart-interval *seconds***]**

Example:

```
ciscoasa(config-router)# nsf ietf restart-interval 80
```

You can specify the length of the graceful restart interval, in seconds. Valid values are from 1 to 1800 seconds. The default value is 120 seconds.

Graceful restart might be terminated when restart interval is configured with a value less than the time taken for the adjacency to come up. For example, a restart interval below 30 seconds, is not supported.

Step 2 Enable IETF NSF helper mode on NSF-aware device:

nsf ietf helper [strict-lsa-checking]

Example:

```
ciscoasa(config-router)# nsf ietf helper
```

The strict-LSA-checking keyword indicates that the helper router will terminate the process of the restarting router if it detects that there is a change to a LSA that would be flooded to the restarting router, or if there is a changed LSA on the retransmission list of the restarting router when the graceful restart process is initiated.

This command is enabled by default. Using the no form of the command disables it.

Configuring Graceful Restart for OSPFv3

Configuring the NSF graceful-restart feature for OSPFv3 involves two steps; configuring a device to be NSF-capable and then configuring a device to be NSF-aware.

Procedure

Step 1 Enable IPv6 processing on an interface that has not been configured with an explicit IPv6 address:

interface physical_interface ipv6 enable

Example:

```
ciscoasa(config)# interface ethernet 0/0
ciscoasa(config-if)# ipv6 enable
```

The physical_interface argument identifies the interface that participates in OSPFv3 NSF.

Step 2 Enable graceful-restart for OSPFv3 on a NSF-capable device:

graceful-restart [restart interval seconds]

Example:

```
ciscoasa(config-router)# graceful-restart restart interval 80
```

The restart interval seconds specifies the length of the graceful restart interval, in seconds. Valid values are from 1 to 1800 seconds. The default value is 120 seconds.

Graceful restart might be terminated when restart interval is configured with a value less than the time taken for the adjacency to come up.For example, a restart interval below 30 seconds, is not supported.

Step 3 Enable graceful-restart for OSPFv3 on a NSF-aware device:

graceful-restart helper [strict-lsa-checking]

Example:

```
ciscoasa(config-router)# graceful-restart helper strict-lsa-checking
```

The strict-LSA-checking keyword indicates that the helper router will terminate the process of the restarting router if it detects that there is a change to a LSA that would be flooded to the restarting router, or if there is a changed LSA on the retransmission list of the restarting router when the graceful restart process is initiated.

The graceful-restart helper mode is enabled by default.

Remove the OSPFv2 Configuration

Remove the OSPFv2 configuration.

Procedure

Remove the entire OSPFv2 configuration that you have enabled.

clear configure router ospf *pid*

Example:

```
ciscoasa(config)# clear configure router ospf 1000
```

After the configuration is cleared, you must reconfigure OSPF using the **router ospf** command.

Remove the OSPFv3 Configuration

Remove the OSPFv3 configuration.

Procedure

Remove the entire OSPFv3 configuration that you have enabled:

clear configure ipv6 router ospf *process-id*

Example:

```
ciscoasa(config)# clear configure ipv6 router ospf 1000
```

After the configuration is cleared, you must reconfigure OSPFv3 using the **ipv6 router ospf** command.

Example for OSPFv2

The following example shows how to enable and configure OSPFv2 with various optional processes:

1. To enable OSPFv2, enter the following commands:

   ```
   ciscoasa(config)# router ospf 2
   ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
   ```

2. (Optional) To redistribute routes from one OSPFv2 process to another OSPFv2 process, enter the following commands:

   ```
   ciscoasa(config)# route-map 1-to-2 permit
   ciscoasa(config-route-map)# match metric 1
   ciscoasa(config-route-map)# set metric 5
   ciscoasa(config-route-map)# set metric-type type-1
   ciscoasa(config-route-map)# router ospf 2
   ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
   ```

3. (Optional) To configure OSPFv2 interface parameters, enter the following commands:

   ```
   ciscoasa(config)# router ospf 2
   ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
   ciscoasa(config-rtr)# interface inside
   ciscoasa(config-interface)# ospf cost 20
   ciscoasa(config-interface)# ospf retransmit-interval 15
   ciscoasa(config-interface)# ospf transmit-delay 10
   ciscoasa(config-interface)# ospf priority 20
   ciscoasa(config-interface)# ospf hello-interval 10
   ciscoasa(config-interface)# ospf dead-interval 40
   ciscoasa(config-interface)# ospf authentication-key cisco
   ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco
   ciscoasa(config-interface)# ospf authentication message-digest
   ```

4. (Optional) To configure OSPFv2 area parameters, enter the following commands:

   ```
   ciscoasa(config)# router ospf 2
   ciscoasa(config-rtr)# area 0 authentication
   ciscoasa(config-rtr)# area 0 authentication message-digest
   ciscoasa(config-rtr)# area 17 stub
   ciscoasa(config-rtr)# area 17 default-cost 20
   ```

5. (Optional) To configure the route calculation timers and show the log neighbor up and down messages, enter the following commands:

   ```
   ciscoasa(config-rtr)# timers spf 10 120
   ciscoasa(config-rtr)# log-adj-changes [detail]
   ```

6. (Optional) To show current OSPFv2 configuration settings, enter the **show ospf** command.

   The following is sample output from the **show ospf** command:

```
ciscoasa(config)# show ospf

Routing Process “ospf 2” with ID 10.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 5. Checksum Sum 0x 209a3
        Number of opaque link LSA 0. Checksum Sum 0x     0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

7. To clear the OSPFv2 configuration, enter the following command:

```
ciscoasa(config)# clear configure router ospf pid
```

Examples for OSPFv3

The following example shows how to enable and configure OSPFv3 at the interface level:

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf 1 area 1
```

The following is sample output from the **show running-config ipv6** command:

```
ciscoasa (config)# show running-config ipv6
ipv6 router ospf 1
 log-adjacency-changes
```

The following is sample output from the **show running-config interface** command:

```
ciscoasa (config-if)# show running-config interface GigabitEthernet3/1
interface GigabitEthernet3/1
 nameif fda
 security-level 100
 ip address 1.1.11.1 255.255.255.0 standby 1.1.11.2
 ipv6 address 9098::10/64 standby 9098::11
 ipv6 enable
 ipv6 ospf 1 area 1
```

The following examples show how to configure OSPFv3-specific interfaces:

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# nameif fda
ciscoasa (config-if)# security-level 100
ciscoasa (config-if)# ip address 10.1.11.1 255.255.255.0 standby 10.1.11.2
ciscoasa (config-if)# ipv6 address 9098::10/64 standby 9098::11
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf cost 900
ciscoasa (config-if)# ipv6 ospf hello-interval 20
ciscoasa (config-if)# ipv6 ospf network broadcast
ciscoasa (config-if)# ipv6 ospf database-filter all out
ciscoasa (config-if)# ipv6 ospf flood-reduction
ciscoasa (config-if)# ipv6 ospf mtu-ignore
ciscoasa (config-if)# ipv6 ospf 1 area 1 instance 100
ciscoasa (config-if)# ipv6 ospf encryption ipsec spi 890 esp null md5
12345678901234567890123456789012

ciscoasa (config)# ipv6 router ospf 1
ciscoasa (config)# area 1 nssa
ciscoasa (config)# distance ospf intra-area 190 inter-area 100 external 100
ciscoasa (config)# timers lsa arrival 900
ciscoasa (config)# timers pacing flood 100
ciscoasa (config)# timers throttle lsa 900 900 900
ciscoasa (config)# passive-interface fda
ciscoasa (config)# log-adjacency-changes
ciscoasa (config)# redistribute connected metric 100 metric-type 1 tag 700
```

For an example of how to configure an OSPFv3 virtual link, see the following URL:

http://www.cisco.com/en/US/tech/tk365/technologies_configuration_example09186a0080b8fd06.shtml

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can also use the information provided to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To monitor or display various OSPFv2 routing statistics, enter one of the following commands:

Command	Purpose
show ospf [*process-id* [*area-id*]]	Displays general information about OSPFv2 routing processes.
show ospf border-routers	Displays the internal OSPFv2 routing table entries to the ABR and ASBR.
show ospf [*process-id* [*area-id*]] database	Displays lists of information related to the OSPFv2 database for a specific router.

Command	Purpose
show ospf flood-list *if-name*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPF v2packet pacing). OSPFv2 update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing, packets might be dropped if either of the following topologies exist: • A fast router is connected to a slower router over a point-to-point link. • During flooding, several neighbors send updates to a single router at the same time. Pacing is also used between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out of an interface. Pacing enables OSPFv2 update and retransmission packets to be sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.
show ospf interface [*if_name*]	Displays OSPFv2-related interface information.
show ospf neighbor [*interface-name*] [*neighbor-id*] [**detail**]	Displays OSPFv2 neighbor information on a per-interface basis.
show ospf request-list *neighbor if_name*	Displays a list of all LSAs requested by a router.
show ospf retransmission-list *neighbor if_name*	Displays a list of all LSAs waiting to be resent.
show ospf [*process-id*] summary-address	Displays a list of all summary address redistribution information configured under an OSPFv2 process.
show ospf [*process-id*] **traffic**	Displays a list of different types of packets being sent or received by a specific OSPFv2 instance.
show ospf [*process-id*] virtual-links	Displays OSPFv2-related virtual links information.
show route cluster	Displays additional OSPFv2 route synchronization information in clustering.

To monitor or display various OSPFv3 routing statistics, enter one of the following commands:

Command	Purpose
show ipv6 ospf [*process-id* [*area-id*]]	Displays general information about OSPFv3 routing processes.

Command	Purpose
show ipv6 ospf [*process-id*] **border-routers**	Displays the internal OSPFv3 routing table entries to the ABR and ASBR.
show ipv6 ospf [*process-id* [*area-id*]] database [**external** | **inter-area prefix** | **inter-area-router** | **network** | **nssa-external** | **router** | **area** | **as** | **ref-lsa** | [*destination-router-id*] [**prefix** *ipv6-prefix*] [*link-state-id*]] [**link** [**interface** *interface-name*] [**adv-router** *router-id*] | **self-originate**] [**internal**] [**database-summary**]	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospf [*process-id* [*area-id*]] **events**	Displays OSPFv3 event information.
show ipv6 ospf [*process-id*] [*area-id*] flood-list *interface-type interface-number*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPFv3 packet pacing). OSPFv3 update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing, packets might be dropped if either of the following topologies exist: • A fast router is connected to a slower router over a point-to-point link. • During flooding, several neighbors send updates to a single router at the same time. Pacing is also used between retransmissions to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out of an interface. Pacing enables OSPFv3 update and retransmission packets to be sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.
show ipv6 ospf [*process-id*] [*area-id*] **interface** [*type number*] [**brief**]	Displays OSPFv3-related interface information.
show ipv6 ospf neighbor [*process-id*] [*area-id*] [*interface-type interface-number*] [*neighbor-id*] [**detail**]	Displays OSPFv3 neighbor information on a per-interface basis.
show ipv6 ospf [*process-id*] [*area-id*] **request-list** [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs requested by a router.
show ipv6 ospf [*process-id*] [*area-id*] retransmission-list [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs waiting to be resent.
show ipv6 ospf statistic [*process-id*] [**detail**]	Displays various OSPFv3 statistics.
show ipv6 ospf [*process-id*] summary-prefix	Displays a list of all summary address redistribution information configured under an OSPFv3 process.

Command	Purpose
show ipv6 ospf [*process-id*] **timers** [**lsa-group | rate-limit**]	Displays OSPFv3 timers information.
show ipv6 ospf [*process-id*] **traffic** [*interface_name*]	Displays OSPFv3 traffic-related statistics.
show ipv6 ospf virtual-links	Displays OSPFv3-related virtual links information.
show ipv6 route cluster [**failover**] [**cluster**] [**interface**] [**ospf**] [**summary**]	Displays the IPv6 routing table sequence number, IPv6 reconvergence timer status, and IPv6 routing entries sequence number in a cluster.

History for OSPF

Table 29: Feature History for OSPF

Feature Name	Platform Releases	Feature Information
OSPF Support	7.0(1)	Support was added for route data, authentication, and redistribution and monitoring of routing information using the Open Shortest Path First (OSPF) routing protocol. We introduced the following command: **route ospf**
Dynamic Routing in Multiple Context Mode	9.0(1)	OSPFv2 routing is supported in multiple context mode.
Clustering	9.0(1)	For OSPFv2 and OSPFv3, bulk synchronization, route synchronization, and Spanned EtherChannel load balancing are supported in the clustering environment. We introduced or modified the following commands: **show route cluster**, **show ipv6 route cluster**, **debug route cluster, router-id cluster-pool**.

Feature Name	Platform Releases	Feature Information
OSPFv3 Support for IPv6	9.0(1)	OSPFv3 routing is supported for IPv6. We introduced or modified the following commands: **ipv6 ospf**, **ipv6 ospf area**, **ipv6 ospf cost, ipv6 ospf database-filter all out, ipv6 ospf dead-interval, ipv6 ospf encryption, ipv6 ospf hello-interval, ipv6 ospf mtu-ignore, ipv6 ospf neighbor, ipv6 ospf network, ipv6 ospf flood-reduction, ipv6 ospf priority, ipv6 ospf retransmit-interval, ipv6 ospf transmit-delay, ipv6 router ospf, ipv6 router ospf area, ipv6 router ospf default, ipv6 router ospf default-information, ipv6 router ospf distance, ipv6 router ospf exit, ipv6 router ospf ignore, ipv6 router ospf log-adjacency-changes, ipv6 router ospf no, ipv6 router ospf passive-interface, ipv6 router ospf redistribute, ipv6 router ospf router-id, ipv6 router ospf summary-prefix, ipv6 router ospf timers, area encryption, area range, area stub, area nssa, area virtual-link, default, default-information originate, distance, ignore lsa mospf, log-adjacency-changes, redistribute, router-id, summary-prefix, timers lsa arrival, timers pacing flood, timers pacing lsa-group, timers pacing retransmission, timers throttle, show ipv6 ospf, show ipv6 ospf border-routers, show ipv6 ospf database, show ipv6 ospf events, show ipv6 ospf flood-list, show ipv6 ospf graceful-restart, show ipv6 ospf interface, show ipv6 ospf neighbor, show ipv6 ospf request-list, show ipv6 ospf retransmission-list, show ipv6 ospf statistic, show ipv6 ospf summary-prefix, show ipv6 ospf timers, show ipv6 ospf traffic, show ipv6 ospf virtual-links, show ospf, show running-config ipv6 router, clear ipv6 ospf, clear configure ipv6 router, debug ospfv3, ipv6 ospf neighbor**.

Feature Name	Platform Releases	Feature Information
OSPF support for Fast Hellos	9.2(1)	OSPF supports the Fast Hello Packets feature, resulting in a configuration that results in faster convergence in an OSPF network. We modified the following command: **ospf dead-interval**
Timers	9.2(1)	New OSPF timers were added; old ones were deprecated. We introduced the following commands: timers lsa arrival, timers pacing, timers throttle We removed the following commands: Timers spf, timers lsa-grouping-pacing
Route filtering using access-list	9.2(1)	Route filtering using ACL is now supported. We introduced the following command: distribute-list
OSPF Monitoring enhancements	9.2(1)	Additional OSPF monitoring information was added. We modified the following commands: show ospf events, show ospf rib, show ospf statistics, show ospf border-routers [detail], show ospf interface brief
OSPF redistribute BGP	9.2(1)	OSPF redistribution feature was added. We added the following command: redistribute bgp
OSPF Support for Non-Stop Forwarding (NSF)	9.3(1)	OSPFv2 and OSPFv3 support for NSF was added. We added the following commands: capability, nsf cisco, nsf cisco helper, nsf ietf, nsf ietf helper, nsf ietf helper strict-lsa-checking, graceful-restart, graceful-restart helper, graceful-restart helper strict-lsa-checking

Feature Name	Platform Releases	Feature Information
		NSF wait timer was added. We added a new command for setting the timer for the NSF restart interval. This command was introduced to ensure the wait interval is not longer that the router dead interval. We introduced the following command: **timers nsf wait <seconds>**

CHAPTER 28

EIGRP

This chapter describes how to configure the Cisco ASA to route data, perform authentication, and redistribute routing information using the Enhanced Interior Gateway Routing Protocol (EIGRP).

About EIGRP

EIGRP is an enhanced version of IGRP developed by Cisco. Unlike IGRP and RIP, EIGRP does not send out periodic route updates. EIGRP updates are sent out only when the network topology changes. Key capabilities that distinguish EIGRP from other routing protocols include fast convergence, support for variable-length subnet mask, support for partial updates, and support for multiple network layer protocols.

A router running EIGRP stores all the neighbor routing tables so that it can quickly adapt to alternate routes. If no appropriate route exists, EIGRP queries its neighbors to discover an alternate route. These queries propagate until an alternate route is found. Its support for variable-length subnet masks permits routes to be automatically summarized on a network number boundary. In addition, EIGRP can be configured to summarize on any bit boundary at any interface. EIGRP does not make periodic updates. Instead, it sends partial updates only when the metric for a route changes. Propagation of partial updates is automatically bounded so that only those routers that need the information are updated. As a result of these two capabilities, EIGRP consumes significantly less bandwidth than IGRP.

Neighbor discovery is the process that the ASA uses to dynamically learn of other routers on directly attached networks. EIGRP routers send out multicast hello packets to announce their presence on the network. When the ASA receives a hello packet from a new neighbor, it sends its topology table to the neighbor with an initialization bit set. When the neighbor receives the topology update with the initialization bit set, the neighbor sends its topology table back to the ASA.

The hello packets are sent out as multicast messages. No response is expected to a hello message. The exception to this is for statically defined neighbors. If you use the **neighbor** command, or configure the Hello Interval in ASDM, to configure a neighbor, the hello messages sent to that neighbor are sent as unicast messages. Routing updates and acknowledgements are sent out as unicast messages.

Once this neighbor relationship is established, routing updates are not exchanged unless there is a change in the network topology. The neighbor relationship is maintained through the hello packets. Each hello packet received from a neighbor includes a hold time. This is the time in which the ASA can expect to receive a hello packet from that neighbor. If the ASA does not receive a hello packet from that neighbor within the hold time advertised by that neighbor, the ASA considers that neighbor to be unavailable.

The EIGRP protocol uses four key algorithm technologies, four key technologies, including neighbor discovery/recovery, Reliable Transport Protocol (RTP), and DUAL, which is important for route computations. DUAL saves all routes to a destination in the topology table, not just the least-cost route. The least-cost route is inserted into the routing table. The other routes remain in the topology table. If the main route fails, another route is chosen from the feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination. The feasibility calculation guarantees that the path is not part of a routing loop.

If a feasible successor is not found in the topology table, a route recomputation must occur. During route recomputation, DUAL queries the EIGRP neighbors for a route, who in turn query their neighbors. Routers that do no have a feasible successor for the route return an unreachable message.

During route recomputation, DUAL marks the route as active. By default, the ASA waits for three minutes to receive a response from its neighbors. If the ASA does not receive a response from a neighbor, the route is marked as stuck-in-active. All routes in the topology table that point to the unresponsive neighbor as a feasibility successor are removed.

Note EIGRP neighbor relationships are not supported through the IPsec tunnel without a GRE tunnel.

Guidelines for EIGRP

Firewall Mode Guidelines

Supported only in routed firewall mode. Transparent firewall mode is not supported.

Cluster Guidelines

EIGRP does not form neighbor relationships with cluster peers in individual interface mode.

IPv6 Guidelines

Does not support IPv6.

Context Guidelines

- EIGRP instances cannot form adjacencies with each other across shared interfaces because, by default, inter-context exchange of multicast traffic is not supported across shared interfaces. However, you can use the static neighbor configuration under EIGRP process configuration under EIGRP process to bring up EIGRP neighbourship on a shared interface.
- Inter-context EIGRP on separate interfaces is supported.

Additional Guidelines

- A maximum of one EIGRP process is supported.
- EIGRP adjacency flap occurs whenever a configuration change is applied which results in modifying the routing information (sent or received) from neighbors especially in distribute lists, offset lists, and changes to summarization. After the routers are synchronized, EIGRP reestablishes the adjacency between neighbors. When an adjacency is torn down and reestablished, all learned routes between the neighbors are erased and the entire synchronization between the neighbors is performed newly with the new distribute list.
- There is no restriction on the maximum number of EIGRP neignbours. However, to prevent unnecessary EIGRP flap, we recommend you to limit the number to 500 per unit.

Configure EIGRP

This section describes how to enable the EIGRP process on your system. After you have enabled EIGRP, see the following sections to learn how to customize the EIGRP process on your system.

Enable EIGRP

You can only enable one EIGRP routing process on the ASA.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more **network** statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Configure Interfaces for EIGRP, on page 736.

Enable EIGRP Stub Routing

You can enable, and configure the ASA as an EIGRP stub router. Stub routing decreases memory and processing requirements on the ASA. As a stub router, the ASA does not need to maintain a complete EIGRP routing table because it forwards all nonlocal traffic to a distribution router. Generally, the distribution router need not send anything more than a default route to the stub router.

Only specified routes are propagated from the stub router to the distribution router. As a stub router, the ASA responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” When the ASA is configured as a stub, it sends a special peer information packet to all neighboring routers to report its status as a stub router. Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router depends on the distribution router to send the correct updates to all peers.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more **network** statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see section Configure Passive Interfaces, on page 738.

Step 3 Configure the stub routing process:

eigrp stub {**receive-only** | [**connected**] [**redistributed**] [**static**] [**summary**]}

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router)# eigrp stub {receive-only | [connected] [redistributed] [static]
[summary]}
```

You must specify which networks are advertised by the stub routing process to the distribution router. Static and connected networks are not automatically redistributed into the stub routing process.

Note A stub routing process does not maintain a full topology table. At a minimum, stub routing needs a default route to a distribution router, which makes the routing decisions.

Customize EIGRP

This section describes how to customize the EIGRP routing.

Define a Network for an EIGRP Routing Process

The Network table lets you specify the networks used by the EIGRP routing process. For an interface to participate in EIGRP routing, it must fall within the range of addresses defined by the network entries. For directly connected and static networks to be advertised, they must also fall within the range of the network entries.

The Network table displays the networks configured for the EIGRP routing process. Each row of the table displays the network address and associated mask configured for the specified EIGRP routing process.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more **network** statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Configure Passive Interfaces, on page 738.

Configure Interfaces for EIGRP

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, you can configure a **network** command that includes the network to which the interface is attached, and use the **passive-interface** command to prevent that interface from sending or receiving EIGRP updates.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more network statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Define a Network for an EIGRP Routing Process, on page 735.

Step 3 Control the sending or receiving of candidate default route information:

no default-information {**in** | **out** | **WORD**}

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router)# no default-information {in | out | WORD}
```

Entering the **no default-information in** command causes the candidate default route bit to be blocked on received routes.

Entering the **no default-information out** command disables the setting of the default route bit in advertised routes.

For more information see, Configure Default Information in EIGRP, on page 746.

Step 4 Enable MD5 authentication of EIGRP packets:

authentication mode eigrp as-num **md5**

Example:

```
ciscoasa(config)# authentication mode eigrp 2 md5
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the ASA. If EIGRP is not enabled or if you enter the wrong number, the ASA returns the following error message:

```
% Asystem(100) specified does not exist
```

For more information see Enable EIGRP Authentication on an Interface, on page 740.

Step 5 Set the delay value:

delay *value*

Example:

```
ciscoasa(config-if)# delay 200
```

The *value* argument entered is in tens of microseconds. To set the delay for 2000 microseconds, enter a *value* of 200.

To view the delay value assigned to an interface, use the **show interface** command.

For more information, see Change the Interface Delay Value, on page 739.

Step 6 Change the hello interval:

hello-interval eigrp as-num seconds

Example:

```
ciscoasa(config)# hello-interval eigrp 2 60
```

For more information see Customize the EIGRP Hello Interval and Hold Time, on page 745.

Step 7 Change the hold time:

hold-time eigrp as-num seconds

Example:

```
ciscoasa(config)# hold-time eigrp 2 60
```

For more information see Customize the EIGRP Hello Interval and Hold Time, on page 745.

Configure Passive Interfaces

You can configure one or more interfaces as passive interfaces. In EIGRP, a passive interface does not send or receive routing updates.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing. You can configure one or more **network** statements with this command:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Define a Network for an EIGRP Routing Process, on page 735.

Step 3 Prevent an interface from sending or receiving EIGRP routing message:

passive-interface {**default** | if-name}

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router)# passive-interface {default}
```

Using the **default** keyword disables EIGRP routing updates on all interfaces. Specifying an interface name, as defined by the **nameif** command, disables EIGRP routing updates on the specified interface. You can use multiple **passive-interface** commands in your EIGRP router configuration.

Configure the Summary Aggregate Addresses on Interfaces

You can configure a summary addresses on a per-interface basis. You need to manually define summary addresses if you want to create summary addresses that do not occur at a network number boundary or if you want to use summary addresses on an ASA with automatic route summarization disabled. If any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

Procedure

Step 1 Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP:

interface phy_if

Example:

```
ciscoasa(config)# interface inside
```

Step 2 Create the summary address:

summary-address eigrp *as-num address mask* [*distance*]

Example:

```
ciscoasa(config-if)# summary-address eigrp 2 address mask [20]
```

By default, EIGRP summary addresses that you define have an administrative distance of 5. You can change this value by specifying the optional *distance* argument in the **summary-address** command.

Change the Interface Delay Value

The interface delay value is used in EIGRP distance calculations. You can modify this value on a per-interface basis.

Procedure

Step 1 Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP:

interface phy_if

Example:

```
ciscoasa(config)# interface inside
```

Step 2 Set a delay value:

delay *value*

Example:

```
ciscoasa(config-if)# delay 200
```

The *value* argument entered is in tens of microseconds. To set the delay for 2000 microseconds, you enter a *value* of 200.

Note To view the delay value assigned to an interface, use the **show interface** command.

Enable EIGRP Authentication on an Interface

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

EIGRP route authentication is configured on a per-interface basis. All EIGRP neighbors on interfaces configured for EIGRP message authentication must be configured with the same authentication mode and key for adjacencies to be established.

Note Before you can enable EIGRP route authentication, you must enable EIGRP.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp *as-num*

Example:

```
ciscoasa(config)# router eigrp 2
```

The as-num argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network *ip-addr* [*mask*]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

- You can configure one or more network statements with this command.
- Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that falls within the defined network participate in the EIGRP routing process.

- If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Configure EIGRP, on page 733.

Step 3 Enter interface configuration mode for the interface on which you are configuring EIGRP message authentication:

interface *phy_if*

Example:

```
ciscoasa(config)# interface inside
```

Step 4 Enable MD5 authentication of EIGRP packets:

authentication mode eigrp *as-num* md5

Example:

```
ciscoasa(config)# authentication mode eigrp 2 md5
```

The as-num argument is the autonomous system number of the EIGRP routing process configured on the ASA. If EIGRP is not enabled or if you enter the wrong number, the ASA returns the following error message:

```
% Asystem(100) specified does not exist
```

Step 5 Configure the key used by the MD5 algorithm:

authentication key eigrp *as-num key* key-id *key-id*

Example:

```
ciscoasa(config)# authentication key eigrp 2 cisco key-id 200
```

- The as-num argument is the autonomous system number of the EIGRP routing process configured on the ASA. If EIGRP is not enabled or if you enter the wrong number, the ASA returns the following error message:

  ```
  % Asystem(100) specified does not exist%
  ```

- The key argument can include up to 16 characters, including alphabets, numbers and special characters.White spaces are not allowed, in the key argument.
- The key-id argument is a number that can range from 0 to 255.

Define an EIGRP Neighbor

EIGRP hello packets are sent as multicast packets. If an EIGRP neighbor is located across a non broadcast network, such as a tunnel, you must manually define that neighbor. When you manually define an EIGRP neighbor, hello packets are sent to that neighbor as unicast messages.

Procedure

Step 1 Create an EIGRP routing process and enters router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Define the static neighbor:

neighbor *ip-addr* **interface** *if_name*

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# neighbor 10.0.0.0 interface interface1
```

The *ip-addr* argument is the IP address of the neighbor.

The *if-name* argument is the name of the interface, as specified by the **nameif** command, through which that neighbor is available. You can define multiple neighbors for an EIGRP routing process.

Redistribute Routes Into EIGRP

You can redistribute routes discovered by RIP and OSPF into the EIGRP routing process. You can also redistribute static and connected routes into the EIGRP routing process. You do not need to redistribute connected routes if they fall within the range of a **network** statement in the EIGRP configuration.

Note For RIP only: Before you begin this procedure, you must create a route map to further define which routes from the specified routing protocol are redistributed in to the RIP routing process.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 (Optional) Specifies the default metrics that should be applied to routes redistributed into the EIGRP routing process:

default-metric bandwidth delay reliability loading mtu

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# default-metric bandwidth delay reliability loading mtu
```

If you do not specify a default metric in the EIGRP router configuration, you must specify the metric values in each **redistribute** command. If you specify the EIGRP metrics in the **redistribute** command and have the **default-metric** command in the EIGRP router configuration, the metrics in the **redistribute** command are used.

Step 3 Redistribute connected routes into the EIGRP routing process:

redistribute connected [**metric** bandwidth delay reliability loading mtu] [**route-map** map_name]

Example:

```
ciscoasa(config-router): redistribute connected [metric bandwidth delay reliability loading
 mtu] [route-map map_name]
```

You must specify the EIGRP metric values in the **redistribute** command if you do not have a **default-metric** command in the EIGRP router configuration.

Step 4 Redistribute static routes into the EIGRP routing process:

redistribute static [**metric** bandwidth delay reliability loading mtu] [**route-map** map_name]

Example:

```
ciscoasa(config-router): redistribute static [metric bandwidth delay
reliability loading mtu] [route-map map_name]
```

Step 5 Redistribute routes from an OSPF routing process into the EIGRP routing process:

redistribute ospf pid [**match** {**internal | external** [**1** | **2**] **| nssa-external** [**1** | **2**]}] [**metric** bandwidth delay reliability loading mtu] [**route-map** map_name]

Example:

```
ciscoasa(config-router): redistribute ospf pid [match {internal | external [1 | 2] |
nssa-external [1 | 2]}] [metric bandwidth delay reliability loading mtu] [route-map map_name]
```

Step 6 Redistribute routes from a RIP routing process into the EIGRP routing process:

redistribute rip [**metric** bandwidth delay reliability load mtu] [**route-map map_name**]

Example:

```
ciscoasa(config-router): redistribute rip [metric bandwidth delay
reliability load mtu] [route-map map_name]
```

Filter Networks in EIGRP

Note Before you begin this process, you must create a standard ACL that defines the routes that you want to advertise. That is, create a standard ACL that defines the routes that you want to filter from sending or receiving updates.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

ciscoasa(config-router)# **network** ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more network statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Configure Interfaces for EIGRP, on page 736.

Step 3 Filter networks sent in EIGRP routing updates:

distribute-list acl **out** [**connected** | **ospf** | **rip** | **static** | **interface** if_name]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router): distribute-list acl out [connected]
```

You can specify an interface to apply the filter to only those updates that are sent by that specific interface.

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

Step 4 Filter networks received in EIGRP routing updates:

distribute-list acl **in** [**interface** if_name]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router): distribute-list acl in [interface interface1]
```

You can specify an interface to apply the filter to only those updates that are received by that interface.

Customize the EIGRP Hello Interval and Hold Time

The ASA periodically sends hello packets to discover neighbors and to learn when neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

The hello packet advertises the ASA hold time. The hold time indicates to EIGRP neighbors the length of time the neighbor should consider the ASA reachable. If the neighbor does not receive a hello packet within the advertised hold time, then the ASA is considered unreachable. By default, the advertised hold time is 15 seconds (three times the hello interval).

Both the hello interval and the advertised hold time are configured on a per-interface basis. We recommend setting the hold time to be at minimum three times the hello interval.

Procedure

Step 1 Enter interface configuration mode for the interface on which you are configuring the hello interval or advertised hold time:

interface phy_if

Example:

```
ciscoasa(config)# interface inside
```

Step 2 Change the hello interval:

hello-interval eigrp as-num seconds

Example:

```
ciscoasa(config)# hello-interval eigrp 2 60
```

Step 3 Change the hold time:

hold-time eigrp as-num seconds

Example:

```
ciscoasa(config)# hold-time eigrp 2 60
```

Disable Automatic Route Summarization

Automatic route summarization is enabled by default. The EIGRP routing process summarizes on network number boundaries. This can cause routing problems if you have noncontiguous networks.

For example, if you have a router with the networks 192.168.1.0, 192.168.2.0, and 192.168.3.0 connected to it, and those networks all participate in EIGRP, the EIGRP routing process creates the summary address 192.168.0.0 for those routes. If an additional router is added to the network with the networks 192.168.10.0 and 192.168.11.0, and those networks participate in EIGRP, they will also be summarized as 192.168.0.0. To prevent the possibility of traffic being routed to the wrong location, you should disable automatic route summarization on the routers creating the conflicting summary addresses.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Disable automatic route summarization:

no auto-summary

Example:

```
ciscoasa(config-router)# no auto-summary
```

Automatic summary addresses have a default administrative distance of 5.

Configure Default Information in EIGRP

You can control the sending and receiving of default route information in EIGRP updates. By default, default routes are sent and accepted. Configuring the ASA to disallow default information to be received causes the candidate default route bit to be blocked on received routes. Configuring the ASA to disallow default information to be sent disables the setting of the default route bit in advertised routes.

Procedure

Step 1 Create an EIGRP routing process and enter router configuration mode for this EIGRP process:

router eigrp as-num

Example:

```
ciscoasa(config)# router eigrp 2
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interfaces and networks that participate in EIGRP routing:

network ip-addr [mask]

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

You can configure one or more network statements with this command.

Directly connected and static networks that fall within the defined network are advertised by the ASA. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to have participate in EIGRP routing, but that is attached to a network that you want advertised, see Configure Interfaces for EIGRP, on page 736.

Step 3 Control the sending or receiving of candidate default route information:

no default-information {**in** | **out** | **WORD**}

Example:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router)# no default-information {in | out | WORD}
```

Note Entering the **no default-information in** command causes the candidate default route bit to be blocked on received routes. Entering the **no default-information out** command disables the setting of the default route bit in advertised routes.

Disable EIGRP Split Horizon

Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks, there may be situations where this behavior is not desired. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

If you disable split horizon on an interface, you must disable it for all routers and access servers on that interface.

To disable EIGRP split horizon, perform the following steps:

Procedure

Step 1 Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP:

interface phy_if

Example:

```
ciscoasa(config)# interface phy_if
```

Step 2 Disable the split horizon:

no split-horizon eigrp *as-number*

Example:

```
ciscoasa(config-if)# no split-horizon eigrp 2
```

Restart the EIGRP Process

You can restart an EIGRP process or clear redistribution or clear counters.

Procedure

Restart an EIGRP process or clear redistribution or clear counters:

clear eigrp pid {**1-65535** | **neighbors** | **topology** | **events)**}

Example:

```
ciscoasa(config)# clear eigrp pid 10 neighbors
```

Monitoring for EIGRP

You can use the following commands to monitor the EIGRP routing process. For examples and descriptions of the command output, see the command reference. Additionally, you can disable the logging of neighbor change messages and neighbor warning messages.

To monitor or disable various EIGRP routing statistics, enter one of the following commands:

- **router-id**

  Displays the router-id for this EIGRP process.

- **show eigrp** [*as-number*] **events** [{*start end*} | **type**]

  Displays the EIGRP event log.

- **show eigrp** [*as-number*] **interfaces** [*if-name*] [**detail**]

  Displays the interfaces participating in EIGRP routing.

- **show eigrp** [as-number] **neighbors** [**detail** | **static**] [*if-name*]

  Displays the EIGRP neighbor table.

- **show eigrp** [*as-number*] **topology** [*ip-addr* [**mask**] | **active** | **all-links** | **pending** | **summary** | **zero-successors**]

  Displays the EIGRP topology table.

- **show eigrp** [*as-number*] **traffic**

  Displays EIGRP traffic statistics.

- **show mfib cluster**

  Displays MFIB information in terms of forwarding entries and interfaces.

- **show route cluster**

  Displays additional route synchronization details for clustering.

- **no eigrp log-neighbor-changes**

  Disables the logging of neighbor change messages. Enter this command in router configuration mode for the EIGRP routing process.

- **no eigrp log-neighbor-warnings**

  Disables the logging of neighbor warning messages.

Example for EIGRP

The following example shows how to enable and configure EIGRP with various optional processes:

Procedure

Step 1 To enable EIGRP, enter the following commands:

```
ciscoasa(config)# router eigrp 2
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

Step 2 To configure an interface from sending or receiving EIGRP routing messages, enter the following command:

```
ciscoasa(config-router)# passive-interface {default}
```

Step 3 To define an EIGRP neighbor, enter the following command:

```
ciscoasa(config-router)# neighbor 10.0.0.0 interface interface1
```

Step 4 To configure the interfaces and networks that participate in EIGRP routing, enter the following command:

```
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
```

Step 5 To change the interface delay value used in EIGRP distance calculations, enter the following commands:

```
ciscoasa(config-router)# exit
ciscoasa(config)# interface phy_if
ciscoasa(config-if)# delay 200
```

History for EIGRP

Table 30: Feature History for EIGRP

Feature Name	Platform Releases	Feature Information
EIGRP Support	7.0(1)	Support was added for routing data, performing authentication, and redistributing and monitoring routing information using the Enhanced Interior Gateway Routing Protocol (EIGRP). We introduced the following command: **route eigrp**.
Dynamic Routing in Multiple Context Mode	9.0(1)	EIGRP routing is supported in multiple context mode.

Feature Name	Platform Releases	Feature Information
Clustering	9.0(1)	For EIGRP, bulk synchronization, route synchronization, and layer 2 load balancing are supported in the clustering environment. We introduced or modified the following commands: **show route cluster**, **debug route cluster, show mfib cluster, debug mfib cluster**.
EIGRP Auto-Summary	9.2(1)	For EIGRP, the Auto-Summary field is now disabled by default.

CHAPTER 29

Multicast Routing

This chapter describes how to configure the Cisco ASA to use the multicast routing protocol.

About Multicast Routing

Multicast routing is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of corporate recipients and homes. Applications that take advantage of multicast routing include videoconferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

Multicast routing protocols deliver source traffic to multiple receivers without adding any additional burden on the source or the receivers while using the least network bandwidth of any competing technology. Multicast packets are replicated in the network by ASA enabled with Protocol Independent Multicast (PIM) and other supporting multicast protocols, which results in the most efficient delivery of data to multiple receivers possible.

The ASA supports both stub multicast routing and PIM multicast routing. However, you cannot configure both concurrently on a single ASA.

Note The UDP and non-UDP transports are both supported for multicast routing. However, the non-UDP transport has no FastPath optimization.

Stub Multicast Routing

Stub multicast routing provides dynamic host registration and facilitates multicast routing. When configured for stub multicast routing, the ASA acts as an IGMP proxy agent. Instead of fully participating in multicast routing, the ASA forwards IGMP messages to an upstream multicast router, which sets up delivery of the multicast data. When configured for stub multicast routing, the ASA cannot be configured for PIM sparse or bidirectional mode. You must enable PIM on the interfaces participating in IGMP stub multicast routing.

The ASA supports both PIM-SM and bidirectional PIM. PIM-SM is a multicast routing protocol that uses the underlying unicast routing information base or a separate multicast-capable routing information base. It builds unidirectional shared trees rooted at a single Rendezvous Point (RP) per multicast group and optionally creates shortest-path trees per multicast source.

PIM Multicast Routing

Bidirectional PIM is a variant of PIM-SM that builds bidirectional shared trees connecting multicast sources and receivers. Bidirectional trees are built using a Designated Forwarder (DF) election process operating on each link of the multicast topology. With the assistance of the DF, multicast data is forwarded from sources to the Rendezvous Point (RP), and therefore along the shared tree to receivers, without requiring source-specific state. The DF election takes place during RP discovery and provides a default route to the RP.

Note If the ASA is the PIM RP, use the untranslated outside address of the ASA as the RP address.

Multicast Group Concept

Multicast is based on the concept of a group. An arbitrary group of receivers expresses an interest in receiving a particular data stream. This group does not have any physical or geographical boundaries—the hosts can be located anywhere on the Internet. Hosts that are interested in receiving data flowing to a particular group must join the group using IGMP. Hosts must be a member of the group to receive the data stream.

Multicast Addresses

Multicast addresses specify an arbitrary group of IP hosts that have joined the group and want to receive traffic sent to this group.

Clustering

Multicast routing supports clustering. In Spanned EtherChannel clustering, the control unit sends all multicast routing packets and data packets until fast-path forwarding is established. After fast-path forwarding is established, data units may forward multicast data packets. All data flows are full flows. Stub forwarding flows are also supported. Because only one unit receives multicast packets in Spanned EtherChannel clustering, redirection to the control unit is common. In Individual Interface clustering, units do not act independently. All data and routing packets are processed and forwarded by the control unit. Data units drop all packets that have been sent.

For more information about clustering, see ASA Cluster, on page 271.

Guidelines for Multicast Routing

Context Mode

Supported in single context mode.

Firewall Mode

Supported only in routed firewall mode. Transparent firewall mode is not supported.

IPv6

Does not support IPv6.

Multicast Group

The range of addresses between 224.0.0.0 and 224.0.0.255 is reserved for the use of routing protocols and other topology discovery or maintenance protocols, such as gateway discovery and group membership reporting. Hence, Internet multicast routing from address range 224.0.0/24 is not supported; IGMP group is not created when enabling multicast routing for the reserved addressess.

Clustering

In clustering, for IGMP and PIM, this feature is only supported on the primary unit.

Additional Guidelines

- You must configure an access control rule on the inbound interface to allow traffic to the multicast host, such as 224.1.2.3. However, you cannot specify a destination interface for the rule, or it cannot be applied to multicast connections during initial connection validation.
- PIM/IGMP Multicast routing is not supported on interfaces in a traffic zone.

Enable Multicast Routing

Enabling multicast routing on the ASA, enables IGMP and PIM on all data interfaces by default, but not on the management interface for most models (see Management Slot/Port Interface, on page 402 for interfaces that do not allow through traffic). IGMP is used to learn whether members of a group are present on directly attached subnets. Hosts join multicast groups by sending IGMP report messages. PIM is used to maintain forwarding tables to forward multicast datagrams.

To enable multicast routing on the management interface, you must explicitly set a multicast boundary on the management interface.

Note Only the UDP transport layer is supported for multicast routing.

The following list shows the maximum number of entries for specific multicast tables. Once these limits are reached, any new entries are discarded.

- MFIB—30,000
- IGMP Groups—30,000
- PIM Routes—72,000

Procedure

Enable multicast routing:

multicast-routing

Example:

```
ciscoasa(config)# multicast-routing
```

The number of entries in the multicast routing tables are limited by the amount of RAM on the ASA.

Customize Multicast Routing

This section describes how to customize multicast routing.

Configure Stub Multicast Routing and Forward IGMP Messages

Note Stub multicast routing is not supported concurrently with PIM sparse and bidirectional modes.

An ASA acting as the gateway to the stub area does not need to participate in PIM sparse mode or bidirectional mode. Instead, you can configure it to act as an IGMP proxy agent and forward IGMP messages from hosts connected on one interface to an upstream multicast router on another interface. To configure the ASA as an IGMP proxy agent, forward the host join and leave messages from the stub area interface to an upstream interface. You must also enable PIM on the interfaces participating in stub mode multicast routing.

Procedure

Configure stub multicast routing and forward IGMP messages:

igmp forward interface if_name

Example:

```
ciscoasa(config-if)# igmp forward interface interface1
```

Configure a Static Multicast Route

Configuring static multicast routes lets you separate multicast traffic from unicast traffic. For example, when a path between a source and destination does not support multicast routing, the solution is to configure two multicast devices with a GRE tunnel between them and to send the multicast packets over the tunnel.

When using PIM, the ASA expects to receive packets on the same interface where it sends unicast packets back to the source. In some cases, such as bypassing a route that does not support multicast routing, you may want unicast packets to take one path and multicast packets to take another.

Static multicast routes are not advertised or redistributed.

Procedure

Step 1 Configure a static multicast route:

mroute src_ip src_mask {input_if_name | *rpf_neighbor*} [distance]

Example:

```
ciscoasa(config)# mroute src_ip src_mask {input_if_name | rpf_neighbor} [distance]
```

Step 2 Configure a static multicast route for a stub area:

mroute src_ip src_mask input_if_name [**dense** output_if_name] [distance]

Example:

```
ciscoasa(config)# mroute src_ip src_mask input_if_name [dense output_if_name] [distance]
```

The **dense** *output_if_name* keyword and argument pair is only supported for stub multicast routing.

Configure IGMP Features

IP hosts use IGMP to report their group memberships to directly-connected multicast routers. IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

This section describes how to configure optional IGMP setting on a per-interface basis.

Disable IGMP on an Interface

You can disable IGMP on specific interfaces. This information is useful if you know that there are no multicast hosts on a specific interface and you want to prevent the ASA from sending host query messages on that interface.

Procedure

Disable IGMP on an interface:

no igmp

Example:

```
ciscoasa(config-if)# no igmp
```

To reenable IGMP on an interface, use the **igmp** command.

Note Only the **no igmp** command appears in the interface configuration.

Configure IGMP Group Membership

You can configure the ASA to be a member of a multicast group. Configuring the ASA to join a multicast group causes upstream routers to maintain multicast routing table information for that group and keep the paths for that group active.

Note If you want to forward multicast packets for a specific group to an interface without the ASA accepting those packets as part of the group, see Configure a Statically Joined IGMP Group, on page 758.

Procedure

Configure the ASA to be a member of a multicast group:

igmp join-group group-address

Example:

```
ciscoasa(config-if)# igmp join-group mcast-group
```

The *group-address* argument is the IP address of the group.

Configure a Statically Joined IGMP Group

Sometimes a group member cannot report its membership in the group because of some configuration, or there may be no members of a group on the network segment. However, you still want multicast traffic for that group to be sent to that network segment. You can have multicast traffic for that group sent to the segment by configuring a statically joined IGMP group.

Enter the **igmp static-group** command. The ASA does not accept the multicast packets, but instead forwards them to the specified interface.

Procedure

Configure the ASA statically to join a multicast group on an interface:

igmp static-group

Example:

```
ciscoasa(config-if)# igmp static-group group-address
```

The *group-address* argument is the IP address of the group.

Control Access to Multicast Groups

You can control access to multicast groups by using access control lists.

Procedure

Step 1 Create a standard ACL for the multicast traffic:

access-list name **standard** [**permit | deny]** ip_addr mask

Example:

```
ciscoasa(config)# access-list acl1 standard permit 192.52.662.25
```

You can create more than one entry for a single ACL. You can use extended or standard ACLs.

The *ip_addr mask* argument is the IP address of the multicast group being permitted or denied.

Step 2 Create an extended ACL:'

access-list name **extended** [**permit | deny]** protocol src_ip_addr src_mask dst_ip_addr dst_mask

Example:

```
ciscoasa(config)# access-list acl2 extended permit protocol
src_ip_addr src_mask dst_ip_addr dst_mask
```

The *dst_ip_addr* argument is the IP address of the multicast group being permitted or denied.

Step 3 Apply the ACL to an interface:

igmp access-group acl

Example:

```
ciscoasa(config-if)# igmp access-group acl
```

The *acl* argument is the name of a standard or extended IP ACL.

Limit the Number of IGMP States on an Interface

You can limit the number of IGMP states resulting from IGMP membership reports on a per-interface basis. Membership reports exceeding the configured limits are not entered in the IGMP cache, and traffic for the excess membership reports is not forwarded.

Procedure

Limit the number of IGMP states on an interface:

igmp limit number

Example:

```
ciscoasa(config-if)# igmp limit 50
```

Valid values range from 0 to 500, with 500 being the default value.

Setting this value to 0 prevents learned groups from being added, but manually defined memberships (using the **igmp join-group** and **igmp static-group** commands) are still permitted. The **no** form of this command restores the default value.

Modify the Query Messages to Multicast Groups

The ASA sends query messages to discover which multicast groups have members on the networks attached to the interfaces. Members respond with IGMP report messages indicating that they want to receive multicast packets for specific groups. Query messages are addressed to the all-systems multicast group, which has an address of 224.0.0.1, with a time-to-live value of 1.

These messages are sent periodically to refresh the membership information stored on the ASA. If the ASA discovers that there are no local members of a multicast group still attached to an interface, it stops forwarding multicast packet for that group to the attached network, and it sends a prune message back to the source of the packets.

By default, the PIM designated router on the subnet is responsible for sending the query messages. By default, they are sent once every 125 seconds.

When changing the query response time, by default, the maximum query response time advertised in IGMP queries is 10 seconds. If the ASA does not receive a response to a host query within this amount of time, it deletes the group.

Note The **igmp query-timeout** and **igmp query-interval** commands require IGMP Version 2.

To change the query interval, query response time, and query timeout value, perform the following steps:

Procedure

Step 1 Set the query interval time in seconds:

igmp query-interval *seconds*

Example:

```
ciscoasa(config-if)# igmp query-interval 30
```

Valid values range from 1 to 3600; 125 is the default value.

If the ASA does not hear a query message on an interface for the specified timeout value (by default, 255 seconds), then the ASA becomes the designated router and starts sending the query messages.

Step 2 Change the timeout value of the query:

igmp query-timeout *seconds*

Example:

```
ciscoasa(config-if)# igmp query-timeout 30
```

Valid values range from 60 to 300; 225 is the default value.

Step 3 Change the maximum query response time:

igmp query-max-response-time *seconds*

Valid values range from 1 to 25; 10 is the default value.

Example:

```
ciscoasa(config-if)# igmp query-max-response-time 20
```

Change the IGMP Version

By default, the ASA runs IGMP Version 2, which enables several additional features such as the **igmp query-timeout** and **igmp query-interval** commands.

All multicast routers on a subnet must support the same version of IGMP. The ASA does not automatically detect Version 1 routers and switch to Version 1. However, a mix of IGMP Version 1 and 2 hosts on the subnet works; the ASA running IGMP Version 2 works correctly when IGMP Version 1 hosts are present.

Procedure

Control the version of IGMP that you want to run on the interface:

igmp version {**1** | **2**}

Example:

```
ciscoasa(config-if)# igmp version 2
```

Configure PIM Features

Routers use PIM to maintain forwarding tables to use for forwarding multicast diagrams. When you enable multicast routing on the ASA, PIM and IGMP are automatically enabled on all interfaces.

Note PIM is not supported with PAT. The PIM protocol does not use ports, and PAT only works with protocols that use ports.

This section describes how to configure optional PIM settings.

Enable and Disable PIM on an Interface

You can enable or disable PIM on specific interfaces.

Procedure

Step 1 Enable or reenable PIM on a specific interface:

pim

Example:

```
ciscoasa(config-if)# pim
```

Step 2 Disable PIM on a specific interface:

no pim

Example:

```
ciscoasa(config-if)# no pim
```

Note Only the **no pim** command appears in the interface configuration.

Configure a Static Rendezvous Point Address

All routers within a common PIM sparse mode or bidir domain require knowledge of the PIM RP address. The address is statically configured using the **pim rp-address** command.

Note The ASA does not support Auto-RP. You must use the **pim rp-address** command to specify the RP address.

You can configure the ASA to serve as RP to more than one group. The group range specified in the ACL determines the PIM RP group mapping. If an ACL is not specified, then the RP for the group is applied to the entire multicast group range (224.0.0.0/4).

Procedure

Enable or reenable PIM on a specific interface:

pim rp-address ip_address [acl] [**bidir**]

The *ip_address* argument is the unicast IP address of the router assigned to be a PIM RP.

The *acl* argument is the name or number of a standard ACL that defines with which multicast groups the RP should be used. Do not use a host ACL with this command.

Excluding the **bidir** keyword causes the groups to operate in PIM sparse mode.

Note The ASA always advertises the bidirectional capability in the PIM hello messages, regardless of the actual bidirectional configuration.

Example:

```
ciscoasa(config)# pim rp-address 10.86.75.23 [acl1] [bidir]
```

Configure the Designated Router Priority

The DR is responsible for sending PIM register, join, and prune messages to the RP. When there is more than one multicast router on a network segment, selecting the DR is based on the DR priority. If multiple devices have the same DR priority, then the device with the highest IP address becomes the DR.

By default, the ASA has a DR priority of 1. You can change this value.

Procedure

Change the designated router priority:

pim dr-priority num

Example:

```
ciscoasa(config-if)# pim dr-priority 500
```

The *num* argument can be any number ranging from 1 to 4294967294.

Configure and Filter PIM Register Messages

When the ASA is acting as an RP, you can restrict specific multicast sources from registering with it to prevent unauthorized sources from registering with the RP. The Request Filter pane lets you define the multicast sources from which the ASA will accept PIM register messages.

Procedure

Configure the ASA to filter PIM register messages:

pim accept-register {**list** acl | **route-map** map-name}

Example:

```
ciscoasa(config)# pim accept-register {list acl1 | route-map map2}
```

In the example, the ASA filters PIM register messages *acl1* and route map *map2*.

Configure PIM Message Intervals

Router query messages are used to select the PIM DR. The PIM DR is responsible for sending router query messages. By default, router query messages are sent every 30 seconds. Additionally, every 60 seconds, the ASA sends PIM join or prune messages.

Procedure

Step 1 Send router query messages:

pim hello-interval seconds

Example:

```
ciscoasa(config-if)# pim hello-interval 60
```

Valid values for the *seconds* argument range from 1 to 3600 seconds.

Step 2 Change the amount of time (in seconds) that the ASA sends PIM join or prune messages:

pim join-prune-interval seconds

Example:

```
ciscoasa(config-if)# pim join-prune-interval 60
```

Valid values for the *seconds* argument range from 10 to 600 seconds.

Filter PIM Neighbors

You can define the routers that can become PIM neighbors. By filtering the routers that can become PIM neighbors, you can do the following:

- Prevent unauthorized routers from becoming PIM neighbors.
- Prevent attached stub routers from participating in PIM.

Procedure

Step 1 Use a standard ACL to define the routers that you want to have participate in PIM:

access-list pim_nbr deny *router-IP_addr PIM neighbor*

Example:

```
ciscoasa(config)# access-list pim_nbr deny 10.1.1.1 255.255.255.255
```

In the example, the following ACL, when used with the **pim neighbor-filter** command, prevents the 10.1.1.1 router from becoming a PIM neighbor.

Step 2 Filter neighbor routers:

pim neighbor-filter pim_nbr

Example:

```
ciscoasa(config)# interface GigabitEthernet0/3
ciscoasa(config-if)# pim neighbor-filter pim_nbr
```

In the example, the 10.1.1.1 router is prevented from becoming a PIM neighbor on interface GigabitEthernet0/3.

Configure a Bidirectional Neighbor Filter

The Bidirectional Neighbor Filter pane shows the PIM bidirectional neighbor filters, if any, that are configured on the ASA. A PIM bidirectional neighbor filter is an ACL that defines the neighbor devices that can participate in the DF election. If a PIM bidirectional neighbor filter is not configured for an interface, then there are no restrictions. If a PIM bidirectional neighbor filter is configured, only those neighbors permitted by the ACL can participate in the DF election process.

When a PIM bidirectional neighbor filter configuration is applied to the ASA, an ACL appears in the running configuration with the name *interface-name*_multicast, in which the *interface-name* is the name of the interface to which the multicast boundary filter is applied. If an ACL with that name already exists, a number is appended to the name (for example, inside_multicast_1). This ACL defines which devices can become PIM neighbors of the ASA.

Bidirectional PIM allows multicast routers to keep reduced state information. All of the multicast routers in a segment must be bidirectionally enabled for bidir to elect a DF.

The PIM bidirectional neighbor filters enable the transition from a sparse-mode-only network to a bidir network by letting you specify the routers that should participate in the DF election, while still allowing all routers to participate in the sparse-mode domain. The bidir-enabled routers can elect a DF from among themselves, even when there are non-bidir routers on the segment. Multicast boundaries on the non-bidir routers prevent PIM messages and data from the bidir groups from leaking in or out of the bidir subset cloud.

When a PIM bidirectional neighbor filter is enabled, the routers that are permitted by the ACL are considered to be bidirectionally capable. Therefore, the following is true:

- If a permitted neighbor does not support bidir, then the DF election does not occur.
- If a denied neighbor supports bidir, then the DF election does not occur.
- If a denied neighbor does not support bidir, the DF election can occur.

Procedure

Step 1 Use a standard ACL to define the routers that you want to have participate in PIM:

access-list pim_nbr deny *router-IP_addr PIM neighbor*

Example:

```
ciscoasa(config)# access-list pim_nbr deny 10.1.1.1 255.255.255.255
```

In the example, the following ACL, when used with the **pim neighbor-filter** command, prevents the 10.1.1.1 router from becoming a PIM neighbor.

Step 2 Filter neighbor routers:

pim bidirectional-neighbor-filter pim_nbr

Example:

```
ciscoasa(config)# interface GigabitEthernet0/3
ciscoasa(config-if)# pim bidirectional neighbor-filter pim_nbr
```

In the example, the 10.1.1.1 router is prevented from becoming a PIM bidirectional neighbor on interface GigabitEthernet0/3.

Configure a Multicast Boundary

Address scoping defines domain boundaries so that domains with RPs that have the same IP address do not leak into each other. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

You can set up an administratively scoped boundary on an interface for multicast group addresses. IANA has designated the multicast address range from 239.0.0.0 to 239.255.255.255 as the administratively scoped addresses. This range of addresses can be reused in domains administered by different organizations. The addresses would be considered local, not globally unique.

A standard ACL defines the range of affected addresses. When a boundary is set up, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

You can configure, examine, and filter Auto-RP discovery and announcement messages at the administratively scoped boundary by entering the **filter-autorp** keyword. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary ACL are removed. An Auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Procedure

Configure a multicast boundary:

multicast boundary acl [**filter-autorp**]

Example:

```
ciscoasa(config-if)# multicast boundary acl1 [filter-autorp]
```

Example for Multicast Routing

The following example shows how to enable and configure multicast routing with various optional processes:

1. Enable multicast routing:

   ```
   ciscoasa(config)# multicast-routing
   ```

2. Configure a static multicast route:

   ```
   ciscoasa(config)# mroute src_ip src_mask {input_if_name | rpf_neighbor} [distance]
   ciscoasa(config)# exit
   ```

3. Configure the ASA to be a member of a multicast group:

   ```
   ciscoasa(config)# interface
   ciscoasa(config-if)# igmp join-group group-address
   ```

History for Multicast Routing

Table 31: Feature History for Multicast Routing

Feature Name	Platform Releases	Feature Information
Multicast routing support	7.0(1)	Support was added for multicast routing data, authentication, and redistribution and monitoring of routing information using the multicast routing protocol. We introduced the **multicast-routing** command.
Clustering support	9.0(1)	Support was added for clustering. We introduced the following commands: **debug mfib cluster**, **show mfib cluster**.

Feature Name	Platform Releases	Feature Information
Protocol Independent Multicast Source-Specific Multicast (PIM-SSM) pass-through support	9.5(1)	Support was added to allow PIM-SSM packets to pass through when multicast routing is enabled, unless the ASA is the Last-Hop Router. This allows greater flexibility in choosing a multicast group while also protecting against different attacks; hosts only receive traffic from explicitly-requested sources. We did not change any commands.

PART VI

AAA Servers and the Local Database

CHAPTER 30

AAA and the Local Database

This chapter describes authentication, authorization, and accounting (AAA, pronounced “triple A”). AAA is a a set of services for controlling access to computer resources, enforcing policies, assessing usage, and providing the information necessary to bill for services. These processes are considered important for effective network management and security.

This chapter also describes how to configure the local database for AAA functionality. For external AAA servers, see the chapter for your server type.

About AAA and the Local Database

This section describes AAA and the local database.

Authentication

Authentication provides a way to identify a user, typically by having the user enter a valid username and valid password before access is granted. The AAA server compares a user's authentication credentials with other user credentials stored in a database. If the credentials match, the user is permitted access to the network. If the credentials do not match, authentication fails and network access is denied.

You can configure the Cisco ASA to authenticate the following items:

- All administrative connections to the ASA, including the following sessions:
    - Telnet
    - SSH
    - Serial console
    - ASDM using HTTPS
    - VPN management access

- The **enable** command
- Network access
- VPN access

Authorization

Authorization is the process of enforcing policies: determining what types of activities, resources, or services a user is permitted to access. After a user is authenticated, that user may be authorized for different types of access or activity.

You can configure the ASA to authorize the following items:

- Management commands
- Network access
- VPN access

Accounting

Accounting measures the resources a user consumes during access, which may include the amount of system time or the amount of data that a user has sent or received during a session. Accounting is carried out through the logging of session statistics and usage information, which is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.

Interaction Between Authentication, Authorization, and Accounting

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

AAA Servers and Server Groups

The AAA server is a network server that is used for access control. Authentication identifies the user. Authorization implements policies that determine which resources and services an authenticated user may access. Accounting keeps track of time and data resources that are used for billing and analysis.

If you want to use an external AAA server, you must first create a AAA server group for the protocol that the external server uses, and add the server to the group. You can create more than one group per protocol, and separate groups for all protocols that you want to use. Each server group is specific to one type of server or service.

See the following topics for details on how to create the groups:

- Configure RADIUS Server Groups, on page 797
- Configure TACACS+ Server Groups, on page 807
- Configure LDAP Server Groups, on page 819
- Configure Kerberos AAA Server Groups, on page 825

- Configure RSA SecurID AAA Server Groups, on page 832

See the VPN configuration guide for more information on using Kerberos Constrained Delegation and HTTP Form.

The following table summarizes the supported types of server and their uses, including the local database.

Table 32: Supported Services for AAA Servers

Server Type and Service	Authentication	Authorization	Accounting
Local Database			
Administrators	Yes	Yes	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	Yes	Yes	No
RADIUS			
Administrators	Yes	Yes	Yes
VPN Users	Yes	Yes	Yes
Firewall Sessions (AAA rules)	Yes	Yes	Yes
TACACS+			
Administrators	Yes	Yes	Yes
VPN Users	Yes	No	Yes
Firewall Sessions (AAA rules)	Yes	Yes	Yes
LDAP			
Administrators	Yes	No	No
VPN Users	Yes	Yes	No
Firewall Sessions (AAA rules)	Yes	No	No
Kerberos			
Administrators	Yes	No	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	Yes	No	No
SDI (RSA SecurID)			
Administrators	Yes	No	No
VPN Users	Yes	No	No

Server Type and Service	Authentication	Authorization	Accounting
Firewall Sessions (AAA rules)	Yes	No	No
HTTP Form			
Administrators	No	No	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	No	No	No
Notes • RADIUS—Accounting for administrators does not include command accounting. • RADIUS—Authorization for firewall sessions is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response. • TACACS+—Accounting for administrators includes command accounting. • HTTP Form—Authentication and SSO operations for clientless SSL VPN user sessions only.			

About the Local Database

The ASA maintains a local database that you can populate with user profiles. You can use a local database instead of AAA servers to provide user authentication, authorization, and accounting.

You can use the local database for the following functions:

- ASDM per-user access
- Console authentication
- Telnet and SSH authentication
- **enable** command authentication

  This setting is for CLI-access only and does not affect the Cisco ASDM login.
- Command authorization

  If you turn on command authorization using the local database, then the Cisco ASA refers to the user privilege level to determine which commands are available. Otherwise, the privilege level is not generally used. By default, all commands are either privilege level 0 or level 15.
- Network access authentication
- VPN client authentication

For multiple context mode, you can configure usernames in the system execution space to provide individual logins at the CLI using the **login** command; however, you cannot configure any AAA rules that use the local database in the system execution space.

Note You cannot use the local database for network access authorization.

Fallback Support

The local database can act as a fallback method for several functions. This behavior is designed to help you prevent accidental lockout from the ASA.

When a user logs in, the servers in the group are accessed one at a time, starting with the first server that you specify in the configuration, until a server responds. If all servers in the group are unavailable, the ASA tries the local database if you have configured it as a fallback method (for management authentication and authorization only). If you do not have a fallback method, the ASA continues to try the AAA servers.

For users who need fallback support, we recommend that their usernames and passwords in the local database match their usernames and passwords on the AAA servers. This practice provides transparent fallback support. Because the user cannot determine whether a AAA server or the local database is providing the service, using usernames and passwords on AAA servers that are different than the usernames and passwords in the local database means that the user cannot be certain which username and password should be given.

The local database supports the following fallback functions:

- Console and enable password authentication—If the servers in the group are all unavailable, the ASA uses the local database to authenticate administrative access, which can also include enable password authentication.
- Command authorization—If the TACACS+ servers in the group are all unavailable, the local database is used to authorize commands based on privilege levels.
- VPN authentication and authorization—VPN authentication and authorization are supported to enable remote access to the ASA if AAA servers that normally support these VPN services are unavailable. When a VPN client of an administrator specifies a tunnel group configured to fallback to the local database, the VPN tunnel can be established even if the AAA server group is unavailable, provided that the local database is configured with the necessary attributes.

How Fallback Works with Multiple Servers in a Group

If you configure multiple servers in a server group and you enable fallback to the local database for the server group, fallback occurs when no server in the group responds to the authentication request from the ASA. To illustrate, consider this scenario:

You configure an LDAP server group with two Active Directory servers, server 1 and server 2, in that order. When the remote user logs in, the ASA attempts to authenticate to server 1.

If server 1 responds with an authentication failure (such as user not found), the ASA does not attempt to authenticate to server 2.

If server 1 does not respond within the timeout period (or the number of authentication attempts exceeds the configured maximum), the ASA tries server 2.

If both servers in the group do not respond, and the ASA is configured to fall back to the local database, the ASA tries to authenticate to the local database.

Guidelines for the Local Database

Make sure that you prevent a lockout from the ASA when using the local database for authentication or authorization.

Add a User Account to the Local Database

To add a user to the local database, perform the following steps:

Procedure

Step 1 Create the user account.

username *username* **password** *password* [**privilege** *priv_level*]

Example:

```
ciscoasa(config)# username exampleuser1 password madmaxfuryroadrules privilege 1
```

The**username** *username* keyword is a string from 3 to 64 characters long, using any combination of ASCII printable characters (character codes 32-126), with the exception of spaces and the question mark. The **password** *password* keyword is a string from 3 to 32 characters long, and can be any combination of ASCII printable characters (character codes 32-126), with the exception of spaces and the question mark. The **privilege** *priv_level* keyword sets the privilege level, which ranges from 0 to 15. The default is 2. This privilege level is used with command authorization.

Caution If you do not use command authorization (the **aaa authorization console LOCAL** command), then the default level 2 allows management access to privileged EXEC mode. If you want to limit access to privileged EXEC mode, either set the privilege level to 0 or 1, or use the **service-type** command.

These less-used options are not shown in the above syntax: The **nopassword** keyword creates a user account that accepts any password; this option is insecure and is not recommended.

The **encrypted** keyword indicates that the password is encrypted (using an MD5-based hash or a PBKDF2 (Password-Based Key Derivation Function 2) hash). When you define a password in the **username** command, the ASA encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **username** command does not show the actual password; it shows the encrypted password followed by the **encrypted** keyword. For example, if you enter the password “test,” the **show running-config** command output would appear as something similar to the following:

```
username user1 password DLaUiAX3l78qgoB5c7iVNw== encrypted
```

The only time you would actually enter the **encrypted** keyword at the CLI is if you are cutting and pasting a configuration file for use in another ASA, and you are using the same password.

Step 2 (Optional) Configure username attributes.

username *username* **attributes**

Example:

```
ciscoasa(config)# username exampleuser1 attributes
```

The *username* argument is the username that you created in the first step.

By default, VPN users that you add with this command have no attributes or group policy association. You must configure all values explicitly using the **username attributes** command. See the VPN configuration guide for more information.

Step 3 (Optional) Configure the user level if you configured management authorization using the **aaa authorization exec** command.

service-type {**admin** | **nas-prompt** | **remote-access**}

Example:

```
ciscoasa(config-username)# service-type admin
```

The **admin** keyword allows full access to any services specified by the **aaa authentication console LOCAL** commands. The **admin** keyword is the default.

The **nas-prompt** keyword allows access to the CLI when you configure the **aaa authentication** {**telnet** | **ssh** | **serial**} **console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you enable authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command (or the **login** command).

The **remote-access** keyword denies management access. You cannot use any services specified by the **aaa authentication console** commands (excluding the **serial** keyword; serial access is allowed).

Step 4 (Optional) For public key authentication for SSH connections to the ASA on a per-user basis, see Configure SSH Access, on page 837.

Step 5 (Optional) If you are using this username for VPN authentication, you can configure many VPN attributes for the user. See the VPN configuration guide for more information.

Examples

The following example assigns a privilege level of 15 to the admin user account:

```
ciscoasa(config)# username admin password farscape1 privilege 15
```

The following example enables management authorization, creates a user account with a password, enters username configuration mode, and specifies a **service-type** of **nas-prompt**:

```
ciscoasa(config)# aaa authorization exec authentication-server
ciscoasa(config)# username user1 password gOrgeOus
ciscoasa(config)# username user1 attributes
ciscoasa(config-username)# service-type nas-prompt
```

Monitoring the Local Database

See the following commands for monitoring the local database:

- **show aaa-server**

  This command shows the configured database statistics. To clear the AAA server statistics, enter the **clear aaa-server statistics** command.

- **show running-config aaa-server**

  This command shows the AAA server running configuration. To clear AAA server configuration, enter the **clear configure aaa-server** command.

History for the Local Database

Table 33: History for the Local Database

Feature Name	Platform Releases	Description
Local database configuration for AAA	7.0(1)	Describes how to configure the local database for AAA use. We introduced the following commands: **username**, **aaa authorization exec authentication-server**, **aaa authentication console LOCAL**, **aaa authorization exec LOCAL**, **service-type, aaa authentication** {**telnet | ssh | serial} console LOCAL**, **aaa authentication http console LOCAL**, **aaa authentication enable console LOCAL**, **show running-config aaa-server**, **show aaa-server**, **clear configure aaa-server**, **clear aaa-server statistics**.
Support for SSH public key authentication	9.1(2)	You can now enable public key authentication for SSH connections to the ASA on a per-user basis. You can specify a public key file (PKF) formatted key or a Base64 key. The PKF key can be up to 4096 bits. Use PKF format for keys that are too large to for the ASA support of the Base64 format (up to 2048 bits). We introduced the following commands: **ssh authentication**. *Also available in 8.4(4.1); PKF key format support is only in 9.1(2).*

CHAPTER 31

RADIUS Servers for AAA

This chapter describes how to configure RADIUS servers for AAA.

About RADIUS Servers for AAA

The Cisco ASA supports the following RFC-compliant RADIUS servers for AAA:

- Cisco Secure ACS 3.2, 4.0, 4.1, 4.2, and 5.x
- Cisco Identity Services Engine (ISE)
- RSA RADIUS in RSA Authentication Manager 5.2, 6.1, and 7.x
- Microsoft

Supported Authentication Methods

The ASA supports the following authentication methods with RADIUS servers:

- PAP—For all connection types.
- CHAP and MS-CHAPv1—For L2TP-over-IPsec connections.
- MS-CHAPv2—For L2TP-over-IPsec connections, and for regular IPsec remote access connections when the password management feature is enabled. You can also use MS-CHAPv2 with clientless connections.
- Authentication Proxy modes—For RADIUS-to Active-Directory, RADIUS-to-RSA/SDI, RADIUS-to-Token server, and RSA/SDI-to-RADIUS connections,

Note

To enable MS-CHAPv2 as the protocol used between the ASA and the RADIUS server for a VPN connection, password management must be enabled in the tunnel group general attributes. Enabling password management generates an MS-CHAPv2 authentication request from the ASA to the RADIUS server. See the description of the **password-management** command for details.

If you use double authentication and enable password management in the tunnel group, then the primary and secondary authentication requests include MS-CHAPv2 request attributes. If a RADIUS server does not support MS-CHAPv2, then you can configure that server to send a non-MS-CHAPv2 authentication request by using the **no mschapv2-capable** command.

User Authorization of VPN Connections

The ASA can use RADIUS servers for user authorization of VPN remote access and firewall cut-through-proxy sessions using dynamic ACLs or ACL names per user. To implement dynamic ACLs, you must configure the RADIUS server to support them. When the user authenticates, the RADIUS server sends a downloadable ACL or ACL name to the ASA. Access to a given service is either permitted or denied by the ACL. The ASA deletes the ACL when the authentication session expires.

In addition to ACLs, the ASA supports many other attributes for authorization and setting of permissions for VPN remote access and firewall cut-through proxy sessions.

Supported Sets of RADIUS Attributes

The ASA supports the following sets of RADIUS attributes:

- Authentication attributes defined in RFC 2138.
- Accounting attributes defined in RFC 2139.
- RADIUS attributes for tunneled protocol support, defined in RFC 2868.
- Cisco IOS Vendor-Specific Attributes (VSAs), identified by RADIUS vendor ID 9.
- Cisco VPN-related VSAs, identified by RADIUS vendor ID 3076.
- Microsoft VSAs, defined in RFC 2548.

Supported RADIUS Authorization Attributes

Authorization refers to the process of enforcing permissions or attributes. A RADIUS server defined as an authentication server enforces permissions or attributes if they are configured. These attributes have vendor ID 3076.

The following table lists the supported RADIUS attributes that can be used for user authorization.

Note RADIUS attribute names do not contain the cVPN3000 prefix. Cisco Secure ACS 4.x supports this new nomenclature, but attribute names in pre-4.0 ACS releases still include the cVPN3000 prefix. The ASAs enforce the RADIUS attributes based on attribute numeric ID, not attribute name.

All attributes listed in the following table are downstream attributes that are sent from the RADIUS server to the ASA except for the following attribute numbers: 146, 150, 151, and 152. These attribute numbers are upstream attributes that are sent from the ASA to the RADIUS server. RADIUS attributes 146 and 150 are sent from the ASA to the RADIUS server for authentication and authorization requests. All four previously listed attributes are sent from the ASA to the RADIUS server for accounting start, interim-update, and stop requests. Upstream RADIUS attributes 146, 150, 151, and 152 were introduced in Version 8.4(3).

Table 34: Supported RADIUS Authorization Attributes

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Access-Hours	Y	1	String	Single	Name of the time range, for example, Business-hours
Access-List-Inbound	Y	86	String	Single	ACL ID
Access-List-Outbound	Y	87	String	Single	ACL ID
Address-Pools	Y	217	String	Single	Name of IP local pool
Allow-Network-Extension-Mode	Y	64	Boolean	Single	0 = Disabled 1 = Enabled
Authenticated-User-Idle-Timeout	Y	50	Integer	Single	1-35791394 minutes
Authorization-DN-Field	Y	67	String	Single	Possible values: UID, OU, O, CN, L, SP, C, EA, T, N, GN, SN, I, GENQ, DNQ, SER, use-entire-name
Authorization-Required		66	Integer	Single	0 = No 1 = Yes
Authorization-Type	Y	65	Integer	Single	0 = None 1 = RADIUS 2 = LDAP
Banner1	Y	15	String	Single	Banner string to display for Cisco VPN remote access sessions: IPsec IKEv1, AnyConnect SSL-TLS/DTLS/IKEv2, and Clientless SSL

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Banner2	Y	36	String	Single	Banner string to display for Cisco VPN remote access sessions: IPsec IKEv1, AnyConnect SSL-TLS/DTLS/IKEv2, and Clientless SSL. The Banner2 string is concatenated to the Banner1 string , if configured.
Cisco-IP-Phone-Bypass	Y	51	Integer	Single	0 = Disabled 1 = Enabled
Cisco-LEAP-Bypass	Y	75	Integer	Single	0 = Disabled 1 = Enabled
Client Type	Y	150	Integer	Single	1 = Cisco VPN Client (IKEv1) 2 = AnyConnect Client SSL VPN 3 = Clientless SSL VPN 4 = Cut-Through-Proxy 5 = L2TP/IPsec SSL VPN 6 = AnyConnect Client IPsec VPN (IKEv2)
Client-Type-Version-Limiting	Y	77	String	Single	IPsec VPN version number string
DHCP-Network-Scope	Y	61	String	Single	IP Address
Extended-Authentication-On-Rekey	Y	122	Integer	Single	0 = Disabled 1 = Enabled

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Framed-Interface-Id	Y	96	String	Single	Assigned IPv6 interface ID. Combines with Framed-IPv6-Prefix to create a complete assigned IPv6 address. For example: Framed-Interface-ID=1:1:1:1 combined with Framed-IPv6-Prefix=2001:0db8::/64 gives the assigned IP address 2001:0db8::1:1:1:1.
Framed-IPv6-Prefix	Y	97	String	Single	Assigned IPv6 prefix and length. Combines with Framed-Interface-Id to create a complete assigned IPv6 address. For example: prefix 2001:0db8::/64 combined with Framed-Interface-Id=1:1:1:1 gives the IP address 2001:0db8::1:1:1:1. You can use this attribute to assign an IP address without using Framed-Interface-Id, by assigning the full IPv6 address with prefix length /128, for example, Framed-IPv6-Prefix=2001:0db8::1/128

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Group-Policy	Y	25	String	Single	Sets the group policy for the remote access VPN session. For Versions 8.2.x and later, use this attribute instead of IETF-Radius-Class. You can use one of the following formats: • *group policy name* • OU=*group policy name* • OU=*group policy name*;
IE-Proxy-Bypass-Local		83	Integer	Single	0 = None 1 = Local
IE-Proxy-Exception-List		82	String	Single	New line (\n) separated list of DNS domains
IE-Proxy-PAC-URL	Y	133	String	Single	PAC address string
IE-Proxy-Server		80	String	Single	IP address
IE-Proxy-Server-Policy		81	Integer	Single	1 = No Modify 2 = No Proxy 3 = Auto detect 4 = Use Concentrator Setting
IKE-KeepAlive-Confidence-Interval	Y	68	Integer	Single	10-300 seconds
IKE-Keepalive-Retry-Interval	Y	84	Integer	Single	2-10 seconds
IKE-Keep-Alives	Y	41	Boolean	Single	0 = Disabled 1 = Enabled
Intercept-DHCP-Configure-Msg	Y	62	Boolean	Single	0 = Disabled 1 = Enabled
IPsec-Allow-Passwd-Store	Y	16	Boolean	Single	0 = Disabled 1 = Enabled

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
IPsec-Authentication		13	Integer	Single	0 = None 1 = RADIUS 2 = LDAP (authorization only) 3 = NT Domain 4 = SDI 5 = Internal 6 = RADIUS with Expiry 7 = Kerberos/Active Directory
IPsec-Auth-On-Rekey	Y	42	Boolean	Single	0 = Disabled 1 = Enabled
IPsec-Backup-Server-List	Y	60	String	Single	Server Addresses (space delimited)
IPsec-Backup-Servers	Y	59	String	Single	1 = Use Client-Configured list 2 = Disable and clear client list 3 = Use Backup Server list
IPsec-Client-Firewall-Filter-Name		57	String	Single	Specifies the name of the filter to be pushed to the client as firewall policy
IPsec-Client-Firewall-Filter-Optional	Y	58	Integer	Single	0 = Required 1 = Optional
IPsec-Default-Domain	Y	28	String	Single	Specifies the single default domain name to send to the client (1-255 characters).
IPsec-IKE-Peer-ID-Check	Y	40	Integer	Single	1 = Required 2 = If supported by peer certificate 3 = Do not check
IPsec-IP-Compression	Y	39	Integer	Single	0 = Disabled 1 = Enabled
IPsec-Mode-Config	Y	31	Boolean	Single	0 = Disabled 1 = Enabled
IPsec-Over-UDP	Y	34	Boolean	Single	0 = Disabled 1 = Enabled

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
IPsec-Over-UDP-Port	Y	35	Integer	Single	4001- 49151. The default is 10000.
IPsec-Required-Client-Firewall-Capability	Y	56	Integer	Single	0 = None 1 = Policy defined by remote FW Are-You-There (AYT) 2 = Policy pushed CPP 4 = Policy from server
IPsec-Sec-Association		12	String	Single	Name of the security association
IPsec-Split-DNS-Names	Y	29	String	Single	Specifies the list of secondary domain names to send to the client (1-255 characters).
IPsec-Split-Tunneling-Policy	Y	55	Integer	Single	0 = No split tunneling 1 = Split tunneling 2 = Local LAN permitted
IPsec-Split-Tunnel-List	Y	27	String	Single	Specifies the name of the network or ACL that describes the split tunnel inclusion list.
IPsec-Tunnel-Type	Y	30	Integer	Single	1 = LAN-to-LAN 2 = Remote access
IPsec-User-Group-Lock		33	Boolean	Single	0 = Disabled 1 = Enabled
IPv6-Address-Pools	Y	218	String	Single	Name of IP local pool-IPv6
IPv6-VPN-Filter	Y	219	String	Single	ACL value
L2TP-Encryption		21	Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bits 4 = 128 bits 8 = Stateless-Req 15= 40/128-Encr/Stateless-Req
L2TP-MPPC-Compression		38	Integer	Single	0 = Disabled 1 = Enabled

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Member-Of	Y	145	String	Single	Comma-delimited string, for example: `Engineering, Sales` An administrative attribute that can be used in dynamic access policies. It does not set a group policy.
MS-Client-Subnet-Mask	Y	63	Boolean	Single	An IP address
NAC-Default-ACL		92	String		ACL
NAC-Enable		89	Integer	Single	0 = No 1 = Yes
NAC-Revalidation-Timer		91	Integer	Single	300-86400 seconds
NAC-Settings	Y	141	String	Single	Name of the NAC policy
NAC-Status-Query-Timer		90	Integer	Single	30-1800 seconds
Perfect-Forward-Secrecy-Enable	Y	88	Boolean	Single	0 = No 1 = Yes
PPTP-Encryption		20	Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bits 4 = 128 bits 8 = Stateless-Required 15= 40/128-Encr/Stateless-Req
PPTP-MPPC-Compression		37	Integer	Single	0 = Disabled 1 = Enabled
Primary-DNS	Y	5	String	Single	An IP address
Primary-WINS	Y	7	String	Single	An IP address
Privilege-Level	Y	220	Integer	Single	An integer between 0 and 15.

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Required-Client-Firewall-Vendor-Code	Y	45	Integer	Single	1 = Cisco Systems (with Cisco Integrated Client) 2 = Zone Labs 3 = NetworkICE 4 = Sygate 5 = Cisco Systems (with Cisco Intrusion Prevention Security Agent)
Required-Client-Firewall-Description	Y	47	String	Single	String
Required-Client-Firewall-Product-Code	Y	46	Integer	Single	Cisco Systems Products: 1 = Cisco Intrusion Prevention Security Agent or Cisco Integrated Client (CIC) Zone Labs Products: 1 = Zone Alarm 2 = Zone AlarmPro 3 = Zone Labs Integrity NetworkICE Product: 1 = BlackIce Defender/Agent Sygate Products: 1 = Personal Firewall 2 = Personal Firewall Pro 3 = Security Agent
Required-Individual-User-Auth	Y	49	Integer	Single	0 = Disabled 1 = Enabled
Require-HW-Client-Auth	Y	48	Boolean	Single	0 = Disabled 1 = Enabled
Secondary-DNS	Y	6	String	Single	An IP address
Secondary-WINS	Y	8	String	Single	An IP address
SEP-Card-Assignment		9	Integer	Single	Not used

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
Session Subtype	Y	152	Integer	Single	0 = None 1 = Clientless 2 = Client 3 = Client Only Session Subtype applies only when the Session Type (151) attribute has the following values: 1, 2, 3, and 4.
Session Type	Y	151	Integer	Single	0 = None 1 = AnyConnect Client SSL VPN 2 = AnyConnect Client IPSec VPN (IKEv2) 3 = Clientless SSL VPN 4 = Clientless Email Proxy 5 = Cisco VPN Client (IKEv1) 6 = IKEv1 LAN-LAN 7 = IKEv2 LAN-LAN 8 = VPN Load Balancing
Simultaneous-Logins	Y	2	Integer	Single	0-2147483647
Smart-Tunnel	Y	136	String	Single	Name of a Smart Tunnel
Smart-Tunnel-Auto	Y	138	Integer	Single	0 = Disabled 1 = Enabled 2 = AutoStart
Smart-Tunnel-Auto-Signon-Enable	Y	139	String	Single	Name of a Smart Tunnel Auto Signon list appended by the domain name
Strip-Realm	Y	135	Boolean	Single	0 = Disabled 1 = Enabled
SVC-Ask	Y	131	String	Single	0 = Disabled 1 = Enabled 3 = Enable default service 5 = Enable default clientless (2 and 4 not used)

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
SVC-Ask-Timeout	Y	132	Integer	Single	5-120 seconds
SVC-DPD-Interval-Client	Y	108	Integer	Single	0 = Off 5-3600 seconds
SVC-DPD-Interval-Gateway	Y	109	Integer	Single	0 = Off) 5-3600 seconds
SVC-DTLS	Y	123	Integer	Single	0 = False 1 = True
SVC-Keepalive	Y	107	Integer	Single	0 = Off 15-600 seconds
SVC-Modules	Y	127	String	Single	String (name of a module)
SVC-MTU	Y	125	Integer	Single	MTU value 256-1406 in bytes
SVC-Profiles	Y	128	String	Single	String (name of a profile)
SVC-Rekey-Time	Y	110	Integer	Single	0 = Disabled 1-10080 minutes
Tunnel Group Name	Y	146	String	Single	1-253 characters
Tunnel-Group-Lock	Y	85	String	Single	Name of the tunnel group or “none”
Tunneling-Protocols	Y	11	Integer	Single	1 = PPTP 2 = L2TP 4 = IPSec (IKEv1) 8 = L2TP/IPSec 16 = WebVPN 32 = SVC 64 = IPsec (IKEv2) 8 and 4 are mutually exclusive. 0 - 11, 16 - 27, 32 - 43, 48 - 59 are legal values.
Use-Client-Address		17	Boolean	Single	0 = Disabled 1 = Enabled
VLAN	Y	140	Integer	Single	0-4094
WebVPN-Access-List	Y	73	String	Single	Access-List name
WebVPN ACL	Y	73	String	Single	Name of a WebVPN ACL on the device
WebVPN-ActiveX-Relay	Y	137	Integer	Single	0 = Disabled Otherwise = Enabled

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
WebVPN-Apply-ACL	Y	102	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Auto-HTTP-Signon	Y	124	String	Single	Reserved
WebVPN-Citrix-Metaframe-Enable	Y	101	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Content-Filter-Parameters	Y	69	Integer	Single	1 = Java ActiveX 2 = Java Script 4 = Image 8 = Cookies in images
WebVPN-Customization	Y	113	String	Single	Name of the customization
WebVPN-Default-Homepage	Y	76	String	Single	A URL such as http://example-example.com
WebVPN-Deny-Message	Y	116	String	Single	Valid string (up to 500 characters)
WebVPN-Download_Max-Size	Y	157	Integer	Single	0x7fffffff
WebVPN-File-Access-Enable	Y	94	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Browsing-Enable	Y	96	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Entry-Enable	Y	95	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Group-based-HTTP/HTTPS-Proxy-Exception-List	Y	78	String	Single	Comma-separated DNS/IP with an optional wildcard (*) (for example *.cisco.com, 192.168.1.*, wwwin.cisco.com)
WebVPN-Hidden-Shares	Y	126	Integer	Single	0 = None 1 = Visible
WebVPN-Home-Page-Use-Smart-Tunnel	Y	228	Boolean	Single	Enabled if clientless home page is to be rendered through Smart Tunnel.
WebVPN-HTML-Filter	Y	69	Bitmap	Single	1 = Java ActiveX 2 = Scripts 4 = Image 8 = Cookies

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
WebVPN-HTTP-Compression	Y	120	Integer	Single	0 = Off 1 = Deflate Compression
WebVPN-HTTP-Proxy-IP-Address	Y	74	String	Single	Comma-separated DNS/IP:port, with http= or https= prefix (for example http=10.10.10.10:80, https=11.11.11.11:443)
WebVPN-Idle-Timeout-Alert-Interval	Y	148	Integer	Single	0-30. 0 = Disabled.
WebVPN-Keepalive-Ignore	Y	121	Integer	Single	0-900
WebVPN-Macro-Substitution	Y	223	String	Single	Unbounded.
WebVPN-Macro-Substitution	Y	224	String	Single	Unbounded.
WebVPN-Port-Forwarding-Enable	Y	97	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-Exchange-Proxy-Enable	Y	98	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-HTTP-Proxy	Y	99	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-List	Y	72	String	Single	Port forwarding list name
WebVPN-Port-Forwarding-Name	Y	79	String	Single	String name (example, "Corporate-Apps"). This text replaces the default string, "Application Access," on the clientless portal home page.
WebVPN-Post-Max-Size	Y	159	Integer	Single	0x7fffffff
WebVPN-Session-Timeout-Alert-Interval	Y	149	Integer	Single	0-30. 0 = Disabled.
WebVPN Smart-Card-Removal-Disconnect	Y	225	Boolean	Single	0 = Disabled 1 = Enabled
WebVPN-Smart-Tunnel	Y	136	String	Single	Name of a Smart Tunnel

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
WebVPN-Smart-Tunnel-Auto-Sign-On	Y	139	String	Single	Name of a Smart Tunnel auto sign-on list appended by the domain name
WebVPN-Smart-Tunnel-Auto-Start	Y	138	Integer	Single	0 = Disabled 1 = Enabled 2 = Auto Start
WebVPN-Smart-Tunnel-Tunnel-Policy	Y	227	String	Single	One of “e networkname,” “i networkname,” or “a,” where networkname is the name of a Smart Tunnel network list, e indicates the tunnel excluded, i indicates the tunnel specified, and a indicates all tunnels.
WebVPN-SSL-VPN-Client-Enable	Y	103	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSL-VPN-Client-Keep-Installation	Y	105	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSL-VPN-Client-Required	Y	104	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSO-Server-Name	Y	114	String	Single	Valid string
WebVPN-Storage-Key	Y	162	String	Single	
WebVPN-Storage-Objects	Y	161	String	Single	
WebVPN-SVC-Keepalive-Frequency	Y	107	Integer	Single	15-600 seconds, 0=Off
WebVPN-SVC-Client-DPD-Frequency	Y	108	Integer	Single	5-3600 seconds, 0=Off
WebVPN-SVC-DTLS-Enable	Y	123	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SVC-DTLS-MTU	Y	125	Integer	Single	MTU value is from 256-1406 bytes.
WebVPN-SVC-Gateway-DPD-Frequency	Y	109	Integer	Single	5-3600 seconds, 0=Off

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
WebVPN-SVC-Rekey-Time	Y	110	Integer	Single	4-10080 minutes, 0=Off
WebVPN-SVC-Rekey-Method	Y	111	Integer	Single	0 (Off), 1 (SSL), 2 (New Tunnel)
WebVPN-SVC-Compression	Y	112	Integer	Single	0 (Off), 1 (Deflate Compression)
WebVPN-UNIX-Group-ID (GID)	Y	222	Integer	Single	Valid UNIX group IDs
WebVPN-UNIX-User-ID (UIDs)	Y	221	Integer	Single	Valid UNIX user IDs
WebVPN-Upload-Max-Size	Y	158	Integer	Single	0x7fffffff
WebVPN-URL-Entry-Enable	Y	93	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-URL-List	Y	71	String	Single	URL list name
WebVPN-User-Storage	Y	160	String	Single	
WebVPN-VDI	Y	163	String	Single	List of settings

Supported IETF RADIUS Authorization Attributes

The following table lists the supported IETF RADIUS attributes.

Table 35: Supported IETF RADIUS Attributes

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
IETF-Radius-Class	Y	25		Single	For Versions 8.2.x and later, we recommend that you use the Group-Policy attribute (VSA 3076, #25): • *group policy name* • OU=*group policy name* • OU=*group policy name*

Attribute Name	ASA	Attr. No.	Syntax/Type	Single or Multi-Valued	Description or Value
IETF-Radius-Filter-Id	Y	11	String	Single	ACL name that is defined on the ASA, which applies only to full tunnel IPsec and SSL VPN clients.
IETF-Radius-Framed-IP-Address	Y	n/a	String	Single	An IP address
IETF-Radius-Framed-IP-Netmask	Y	n/a	String	Single	An IP address mask
IETF-Radius-Idle-Timeout	Y	28	Integer	Single	Seconds
IETF-Radius-Service-Type	Y	6	Integer	Single	Seconds. Possible Service Type values: • .Administrative—User is allowed access to the configure prompt. • .NAS-Prompt—User is allowed access to the exec prompt. • .remote-access—User is allowed network access
IETF-Radius-Session-Timeout	Y	27	Integer	Single	Seconds

RADIUS Accounting Disconnect Reason Codes

These codes are returned if the ASA encounters a disconnect when sending packets:

Disconnect Reason Code
ACCT_DISC_USER_REQ = 1
ACCT_DISC_LOST_CARRIER = 2
ACCT_DISC_LOST_SERVICE = 3
ACCT_DISC_IDLE_TIMEOUT = 4
ACCT_DISC_SESS_TIMEOUT = 5
ACCT_DISC_ADMIN_RESET = 6

Disconnect Reason Code
ACCT_DISC_ADMIN_REBOOT = 7
ACCT_DISC_PORT_ERROR = 8
ACCT_DISC_NAS_ERROR = 9
ACCT_DISC_NAS_REQUEST = 10
ACCT_DISC_NAS_REBOOT = 11
ACCT_DISC_PORT_UNNEEDED = 12
ACCT_DISC_PORT_PREEMPTED = 13
ACCT_DISC_PORT_SUSPENDED = 14
ACCT_DISC_SERV_UNAVAIL = 15
ACCT_DISC_CALLBACK = 16
ACCT_DISC_USER_ERROR = 17
ACCT_DISC_HOST_REQUEST = 18
ACCT_DISC_ADMIN_SHUTDOWN = 19
ACCT_DISC_SA_EXPIRED = 21
ACCT_DISC_MAX_REASONS = 22

Guidelines for RADIUS Servers for AAA

This section describes the guidelines and limitations that you should check before configuring RADIUS servers for AAA.

- You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode.
- Each group can have up to 16 servers in single mode or 4 servers in multiple mode.

IPv6

The AAA server must use an IPv4 address, but endpoints can use IPv6.

Configure RADIUS Servers for AAA

This section describes how to configure RADIUS servers for AAA.

Procedure

Step 1 Load the ASA attributes into the RADIUS server. The method that you use to load the attributes depends on which type of RADIUS server that you are using:

- If you are using Cisco ACS: the server already has these attributes integrated. You can skip this step.
- For RADIUS servers from other vendors (for example, Microsoft Internet Authentication Service): you must manually define each ASA attribute. To define an attribute, use the attribute name or number, type, value, and vendor code (3076).

Step 2 Configure RADIUS Server Groups, on page 797.

Step 3 Add a RADIUS Server to a Group, on page 800.

Configure RADIUS Server Groups

If you want to use an external RADIUS server for authentication, authorization, or accounting, you must first create at least one RADIUS server group per AAA protocol and add one or more servers to each group.

Procedure

Step 1 Create the RADIUS AAA server group.

aaa-server *group_name* **protocol radius**

Example:

```
ciscoasa(config)# aaa-server servergroup1 protocol radius
ciscoasa(config-aaa-server-group)#
```

When you enter the **aaa-server protocol** command, you enter aaa-server group configuration mode.

Step 2 (Optional.) Specify the maximum number of failed AAA transactions with a RADIUS server in the group before trying the next server.

max-failed-attempts *number*

The range is from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, or their responses are invalid, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

Step 3 (Optional.) Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Where:

- **depletion** [**deadtime** *minutes*] reactivates failed servers only after all of the servers in the group are inactive. This is the default reactivation mode. You can specify the amount of time, between 0 and 1440 minutes, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.
- **timed** reactivates failed servers after 30 seconds of down time.

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode deadtime 20
```

Step 4 (Optional.) Send accounting messages to all servers in the group.

accounting-mode simultaneous

To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.

Example:

```
ciscoasa(config-aaa-server-group)# accounting-mode simultaneous
```

Step 5 (Optional.) Enable the periodic generation of RADIUS interim-accounting-update messages.

interim-accounting-update [**periodic** [*hours*]]

ISE maintains a directory of active sessions based on the accounting records that it receives from NAS devices like the ASA. However, if ISE does not receive any indication that the session is still active (accounting message or posture transactions) for a period of 5 days, it will remove the session record from its database. To ensure that long-lived VPN connections are not removed, configure the group to send periodic interim-accounting-update messages to ISE for all active sessions.

- **periodic** [*hours*] enables the periodic generation and transmission of accounting records for every VPN session that is configured to send accounting records to the server group in question. You can optionally include the interval, in hours, for sending these updates. The default is 24 hours, the range is 1 to 120.
- (No parameters.) If you use this command without the **periodic** keyword, the ASA sends interim-accounting-update messages only when a VPN tunnel connection is added to a clientless VPN session. When this happens the accounting update is generated in order to inform the RADIUS server of the newly assigned IP address.

Example:

```
hostname(config-aaa-server-group)# interim-accounting-update periodic 12
```

Step 6 (Optional.) Enable the RADIUS Dynamic Authorization (ISE Change of Authorization, CoA) services for the AAA server group.

dynamic-authorization [**port** *number*]

Specifying a port is optional. The default is 1700, the range is 1024 to 65535.

When you use the server group in a VPN tunnel, the RADIUS server group will be registered for CoA notification and the ASA will listen to the port for the CoA policy updates from ISE. Enable dynamic authorization only if you are using this server group in a remote access VPN in conjunction with ISE.

Example:

```
ciscoasa(config-aaa-server-group)# dynamic-authorization
```

Step 7 (Optional.) If you do not want to use ISE for authentication, enable authorize-only mode for the RADIUS server group. (Enable authorize-only mode only if you are using this server group in a remote access VPN in conjunction with ISE.)

authorize-only

This indicates that when this server group is used for authorization, the RADIUS Access Request message will be built as an “Authorize Only” request as opposed to the configured password methods defined for the AAA server. If you do configure a common password using **radius-common-pw** command for the RADIUS server, it will be ignored.

For example, you would use authorize-only mode if you want to use certificates for authentication rather than this server group. You would still use this server group for authorization and accounting in the VPN tunnel.

Example:

```
ciscoasa(config-aaa-server-group)# authorize-only
```

Step 8 (Optional.) Merge a downloadable ACL with the ACL received in the Cisco AV pair from a RADIUS packet.

merge-dacl {**before-avpair** | **after-avpair**}

Example:

```
ciscoasa(config-aaa-server-group)# merge-dacl before-avpair
```

This option applies only to VPN connections. For VPN users, ACLs can be in the form of Cisco AV pair ACLs, downloadable ACLs, and an ACL that is configured on the ASA. This option determines whether or not the downloadable ACL and the AV pair ACL are merged, and does not apply to any ACLs configured on the ASA.

The default setting is **no merge dacl**, which specifies that downloadable ACLs will not be merged with Cisco AV pair ACLs. If both an AV pair and a downloadable ACL are received, the AV pair has priority and is used.

The **before-avpair** option specifies that the downloadable ACL entries should be placed before the Cisco AV pair entries.

The **after-avpair** option specifies that the downloadable ACL entries should be placed after the Cisco AV pair entries.

Examples

The following example shows how to add one RADIUS group with a single server:

```
ciscoasa(config)# aaa-server AuthOutbound protocol radius
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
ciscoasa(config-aaa-server-host)# key RadUauthKey
ciscoasa(config-aaa-server-host)# exit
```

The following example shows how to configure an ISE server group for dynamic authorization (CoA) updates and hourly periodic accounting. Included is the tunnel group configuration that configures password authentication with ISE.

```
ciscoasa(config)# aaa-server ise protocol radius
ciscoasa(config-aaa-server-group)# interim-accounting-update periodic 1
ciscoasa(config-aaa-server-group)# dynamic-authorization
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server ise (inside) host 10.1.1.3
ciscoasa(config-aaa-server-host)# key sharedsecret
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# tunnel-group aaa-coa general-attributes
ciscoasa(config-tunnel-general)# address-pool vpn
ciscoasa(config-tunnel-general)# authentication-server-group ise
ciscoasa(config-tunnel-general)# accounting-server-group ise
ciscoasa(config-tunnel-general)# exit
```

The following example shows how to configure a tunnel group for local certificate validation and authorization with ISE. Include the authorize-only command in the server group configuration, because the server group will not be used for authentication.

```
ciscoasa(config)# aaa-server ise protocol radius
ciscoasa(config-aaa-server-group)# authorize-only
ciscoasa(config-aaa-server-group)# interim-accounting-update periodic 1
ciscoasa(config-aaa-server-group)# dynamic-authorization
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server ise (inside) host 10.1.1.3
ciscoasa(config-aaa-server-host)# key sharedsecret
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# tunnel-group aaa-coa general-attributes
ciscoasa(config-tunnel-general)# address-pool vpn
ciscoasa(config-tunnel-general)# authentication certificate
ciscoasa(config-tunnel-general)# authorization-server-group ise
ciscoasa(config-tunnel-general)# accounting-server-group ise
ciscoasa(config-tunnel-general)# exit
```

Add a RADIUS Server to a Group

To add a RADIUS server to a group, perform the following steps:

Procedure

Step 1 Identify the RADIUS server and the AAA server group to which it belongs.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config-aaa-server-group)# aaa-server servergroup1 outside host 10.10.1.1
```

If you do not specify an **(***interface_name***)**, then the ASA uses the **inside** interface by default.

Step 2 Specify how the ASA treats netmasks received in a downloadable ACL from a RADIUS server.

acl-netmask-convert {**auto-detect** | **standard** | **wildcard**}

Example:

```
ciscoasa(config-aaa-server-host)# acl-netmask-convert standard
```

The**auto-detect** keyword specifies that the ASA should attempt to determine the type of netmask expression used. If the ASA detects a wildcard netmask expression, it converts it to a standard netmask expression.

The **standard** keyword specifies that the ASA assumes downloadable ACLs received from the RADIUS server contain only standard netmask expressions. No translation from wildcard netmask expressions is performed.

The **wildcard** keyword specifies that the ASA assumes downloadable ACLs received from the RADIUS server contain only wildcard netmask expressions and converts them all to standard netmask expressions when the ACLs are downloaded.

Step 3 Specify a common password to be used for all users who are accessing a RADIUS authorization server through the ASA.

radius-common-pw *string*

Example:

```
ciscoasa(config-aaa-server-host)# radius-common-pw examplepassword123abc
```

The *string* argument is a case-sensitive, alphanumeric keyword of up to 127 characters to be used as a common password for all authorization transactions with the RADIUS server.

Step 4 Enable MS-CHAPv2 authentication requests to the RADIUS server.

mschapv2-capable

Example:

```
ciscoasa(config-aaa-server-host)# mschapv2-capable
```

Step 5 Specify the timeout value for connection attempts to the server.

timeout *seconds*

Specify the timeout interval (1-300 seconds) for the server; the default is 10 seconds. For each AAA transaction the ASA retries connection attempts (based on the interval defined on the **retry-interval** command) until the timeout is reached. If the number of consecutive failed transactions reaches the limit specified on the max-failed-attempts command in the AAA server group, the AAA server is deactivated and the ASA starts sending requests to another AAA server if it is configured.

Example:

```
ciscoasa(config-aaa-server-host)# timeout 15
```

Step 6 Configure the amount of time between retry attempts for a particular AAA server designated in a previous command.

retry-interval *seconds*

Example:

```
ciscoasa(config-aaa-server-host)# retry-interval 8
```

The *seconds* argument specifies the retry interval (1-10 seconds) for the request. This is the time that the ASA waits before retrying a connection request.

Note For the RADIUS protocol, if the server responds with an ICMP Port Unreachable message, the retry-interval setting is ignored and the AAA server is immediately moved to the failed state. If this is the only server in the AAA group, it is reactivated and another request is sent to it. This is the intended behavior.

Step 7 Send accounting messages to all servers in the group.

accounting-mode simultaneous

Example:

```
ciscoasa(config-aaa-server-group)# accounting-mode simultaneous
```

Enter the **accounting-mode single** command to restore the default of sending messages only to the active server.

Step 8 Specify the authentication port as port number1645, or the server port to be used for authentication of users.

authentication-port *port*

Example:

```
ciscoasa(config-aaa-server-host)# authentication-port 1646
```

Step 9 Specify the accounting port as port number 1646, or the server port to be used for accounting for this host.

accounting-port *port*

Example:

```
ciscoasa(config-aaa-server-host)# accounting-port 1646
```

Step 10 Specify the server secret value used to authenticate the RADIUS server to the ASA. The server secret that you configure should match the one configured on the RADIUS server. If you do not know the server secret value, ask the RADIUS server administrator. The maximum length is 64 characters.

key

Example:

```
ciscoasa(config-aaa-host)# key myexamplekey1
```

The server secret that you configure should match the one configured on the RADIUS server. If you do not know the server secret value, ask the RADIUS server administrator. The maximum length is 64 characters.

Example

The following example shows how to add a RADIUS server to an existing RADIUS server group:

```
ciscoasa(config)# aaa-server svrgrp1 protocol radius
ciscoasa(config-aaa-server-group)# aaa-server svrgrp1 host 192.168.3.4
ciscoasa(config-aaa-server-host)# acl-netmask-convert wildcard
ciscoasa(config-aaa-server-host)# radius-common-pw myexaplepasswordabc123
ciscoasa(config-aaa-server-host)# mschapv2-capable
ciscoasa(config-aaa-server-host)# timeout 9
ciscoasa(config-aaa-server-host)# retry-interval 7
ciscoasa(config-aaa-server-host)# accounting-mode simultaneous
ciscoasa(config-aaa-server-host)# authentication-port 1650
ciscoasa(config-aaa-server-host)# authorization-port 1645
ciscoasa(config-aaa-server-host)# key mysecretkeyexampleiceage2
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)#
```

Monitoring RADIUS Servers for AAA

See the following commands for monitoring the status of RADIUS servers for AAA:

- **show aaa-server**

  This command shows the configured RADIUS server statistics. You can use the **clear aaa-server statistics** command to reset the counters to zero.

- **show running-config aaa-server**

  This command shows the RADIUS server running configuration.

History for RADIUS Servers for AAA

Table 36: History for RADIUS Servers for AAA

Feature Name	Platform Releases	Description
RADIUS Servers for AAA	7.0(1)	Describes how to configure RADIUS servers for AAA. We introduced the following commands: **aaa-server protocol**, **max-failed-attempts**, **reactivation-mode**, **accounting-mode simultaneous**, **aaa-server host**, **show aaa-server**, **show running-config aaa-server**, **clear aaa-server statistics**, **authentication-port**, **accounting-port**, **retry-interval**, **acl-netmask-convert**, **clear configure aaa-server**, **merge-dacl**, **radius-common-pw, key**.
Key vendor-specific attributes (VSAs) sent in RADIUS access request and accounting request packets from the ASA	8.4(3)	Four New VSAs—Tunnel Group Name (146) and Client Type (150) are sent in RADIUS access request packets from the ASA. Session Type (151) and Session Subtype (152) are sent in RADIUS accounting request packets from the ASA. All four attributes are sent for all accounting request packet types: Start, Interim-Update, and Stop. The RADIUS server (for example, ACS and ISE) can then enforce authorization and policy attributes or use them for accounting and billing purposes.

CHAPTER 32

TACACS+ Servers for AAA

This chapter describes how to configure TACACS+ servers used in AAA.

About TACACS+ Servers for AAA

The ASA supports TACACS+ server authentication with the following protocols: ASCII, PAP, CHAP, and MS-CHAPv1.

TACACS+ Attributes

The Cisco ASA provides support for TACACS+ attributes. TACACS+ attributes separate the functions of authentication, authorization, and accounting. The protocol supports two types of attributes: mandatory and optional. Both the server and client must understand a mandatory attribute, and the mandatory attribute must be applied to the user. An optional attribute may or may not be understood or used.

Note To use TACACS+ attributes, make sure that you have enabled AAA services on the NAS.

The following table lists supported TACACS+ authorization response attributes for cut-through-proxy connections.

Table 37: Supported TACACS+ Authorization Response Attributes

Attribute	Description
acl	Identifies a locally configured ACL to be applied to the connection.

Attribute	Description
idletime	Indicates the amount of inactivity in minutes that is allowed before the authenticated user session is terminated.
timeout	Specifies the absolute amount of time in minutes that authentication credentials remain active before the authenticated user session is terminated.

The following table lists supported TACACS+ accounting attributes.

.

Table 38: Supported TACACS+ Accounting Attributes

Attribute	Description
bytes_in	Specifies the number of input bytes transferred during this connection (stop records only).
bytes_out	Specifies the number of output bytes transferred during this connection (stop records only).
cmd	Defines the command executed (command accounting only).
disc-cause	Indicates the numeric code that identifies the reason for disconnecting (stop records only).
elapsed_time	Defines the elapsed time in seconds for the connection (stop records only).
foreign_ip	Specifies the IP address of the client for tunnel connections. Defines the address on the lowest security interface for cut-through-proxy connections.
local_ip	Specifies the IP address that the client connected to for tunnel connections. Defines the address on the highest security interface for cut-through-proxy connections.
NAS port	Contains a session ID for the connection.
packs_in	Specifies the number of input packets transferred during this connection.
packs_out	Specifies the number of output packets transferred during this connection.
priv-level	Set to the user privilege level for command accounting requests or to 1 otherwise.
rem_iddr	Indicates the IP address of the client.

Attribute	Description
service	Specifies the service used. Always set to “shell” for command accounting only.
task_id	Specifies a unique task ID for the accounting transaction.
username	Indicates the name of the user.

Guidelines for TACACS+ Servers for AAA

This section describes the guidelines and limitation that you should check before configuring TACACS+ servers for AAA.

IPv6

The AAA server must use an IPv4 address, but endpoints can use IPv6.

Additional Guidelines

- You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode.
- Each group can have up to 16 servers in single mode or 4 servers in multiple mode.

Configure TACACS+ Servers

This section describes how to configure TACACS+ servers.

Procedure

Step 1 Configure TACACS+ Server Groups, on page 807.

Step 2 Add a TACACS+ Server to a Group, on page 809.

Configure TACACS+ Server Groups

If you want to use a TACACS+ server for authentication, authorization, or accounting, you must first create at least one TACACS+ server group and add one or more servers to each group. You identify TACACS+ server groups by name.

To add a TACACS+ server group, perform the following steps:

Procedure

Step 1 Identify the server group name and the protocol.

aaa-server *server_tag* **protocol tacacs+**

Example:

```
ciscoasa(config)# aaa-server servergroup1 protocol tacacs+
```

When you enter the **aaa-server protocol** command, you enter aaa-server group configuration mode.

Step 2 Specify the maximum number of failed AAA transactions with a AAA server in the group before trying the next server.

max-failed-attempts *number*

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

The *number* argument can range from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, or their responses are invalid, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

Step 3 Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode depletion deadtime 20
```

The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive.

The **deadtime** *minutes* keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

Step 4 Send accounting messages to all servers in the group.

accounting-mode simultaneous

Example:

```
ciscoasa(config-aaa-server-group)# accounting-mode simultaneous
```

To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.

Example

The following example shows how to add one TACACS+ group with one primary and one backup server:

```
ciscoasa(config)# aaa-server AuthInbound protocol tacacs+
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
ciscoasa(config-aaa-server-group)# reactivation-mode depletion deadtime 20
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthInbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# aaa-server AuthInbound (inside) host 10.1.1.2
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey2
ciscoasa(config-aaa-server-host)# exit
```

Add a TACACS+ Server to a Group

To add a TACACS+ server to a group, perform the following steps:

Procedure

Step 1 Identify the TACACS+ server and the server group to which it belongs.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config-aaa-server-group)# aaa-server servergroup1 outside host 10.10.1.1
```

If you do not specify an **(***interface_name***)**, then the ASA uses the **inside** interface by default.

Step 2 Specify the timeout value for connection attempts to the server.

timeout *seconds*

Specify the timeout interval (1-300 seconds) for the server; the default is 10 seconds. For each AAA transaction the ASA retries connection attempts (based on the interval defined on the **retry-interval** command) until the timeout is reached. If the number of consecutive failed transactions reaches the limit specified on the max-failed-attempts command in the AAA server group, the AAA server is deactivated and the ASA starts sending requests to another AAA server if it is configured.

Example:

```
ciscoasa(config-aaa-server-host)# timeout 15
```

Step 3 Specify the server port as port number 49, or the TCP port number used by the ASA to communicate with the TACACS+ server.

server-port *port_number*

Example:

```
ciscoasa(config-aaa-server-host)# server-port 49
```

Step 4 Specify the server secret value used to authenticate the NAS to the TACACS+ server.

key

Example:

```
ciscoasa(config-aaa-host)# key myexamplekey1
```

This value is a case-sensitive, alphanumeric keyword of up to 127 characters, which is the same value as the key on the TACACS+ server. Any characters over 127 are ignored. The key is used between the client and the server to encrypt data between them and must be the same on both the client and server systems. The key cannot contain spaces, but other special characters are allowed.

Monitoring TACACS+ Servers for AAA

See the following commands for monitoring TACACS+ servers for AAA:

- **show aaa-server**

  This command shows the configured TACACS+ server statistics. Enter the **clear aaa-server statistics** command to clear the TACACS+ server statistics.

- **show running-config aaa-server**

  This command shows the TACACS+ server running configuration. Enter the **clear configure aaa-server**command to clear the TACACS+ server configuration.

History for TACACS+ Servers for AAA

Table 39: History for TACACS+ Servers for AAA

Feature Name	Platform Releases	Description
TACACS+ Servers	7.0(1)	Describes how to configure TACACS+ servers for AAA. We introduced the following commands: **aaa-server protocol**, **max-failed-attempts, reactivation-mode, accounting-mode simultaneous, aaa-server host, aaa authorization exec authentication-server**, **server-port, key**, **clear aaa-server statistics**, **clear configure aaa-server**, **show aaa-server**, **show running-config aaa-server**, **username**, **service-type**, **timeout**.

CHAPTER 33

LDAP Servers for AAA

This chapter describes how to configure LDAP servers used in AAA.

About LDAP and the ASA

The Cisco ASA is compatible with the most LDAPv3 directory servers, including:

- Sun Microsystems JAVA System Directory Server, now part of Oracle Directory Server Enterprise Edition, and formerly named the Sun ONE Directory Server
- Microsoft Active Directory
- Novell
- OpenLDAP

By default, the ASA autodetects whether it is connected to Microsoft Active Directory, Sun LDAP, Novell, OpenLDAP, or a generic LDAPv3 directory server. However, if autodetection fails to determine the LDAP server type, you can manually configure it.

How Authentication Works with LDAP

During authentication, the ASA acts as a client proxy to the LDAP server for the user, and authenticates to the LDAP server in either plain text or by using the SASL protocol. By default, the ASA passes authentication parameters, usually a username and password, to the LDAP server in plain text.

The ASA supports the following SASL mechanisms, listed in order of increasing strength:

- Digest-MD5—The ASA responds to the LDAP server with an MD5 value computed from the username and password.
- Kerberos—The ASA responds to the LDAP server by sending the username and realm using the GSSAPI Kerberos mechanism.

The ASA and LDAP server supports any combination of these SASL mechanisms. If you configure multiple mechanisms, the ASA retrieves the list of SASL mechanisms that are configured on the server, and sets the authentication mechanism to the strongest one configured on both the ASA and the server. For example, if both the LDAP server and the ASA support both mechanisms, the ASA selects Kerberos, the stronger of the two.

When user LDAP authentication has succeeded, the LDAP server returns the attributes for the authenticated user. For VPN authentication, these attributes generally include authorization data that is applied to the VPN session. In this case, using LDAP accomplishes authentication and authorization in a single step.

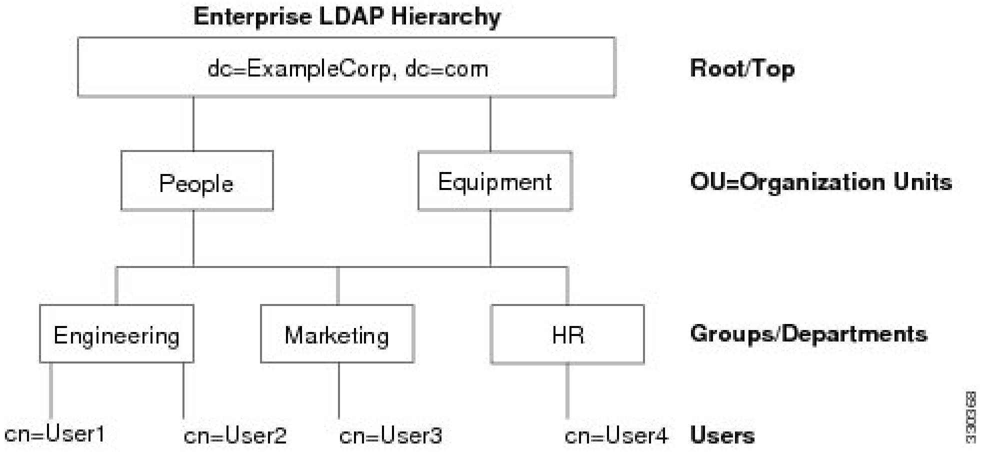
Note For more information about LDAP, see RFCs 1777, 2251, and 2849.

LDAP Hierarchy

Your LDAP configuration should reflect the logical hierarchy of your organization. For example, suppose an employee at your company, Example Corporation, is named Employee1. Employee1 works in the Engineering group. Your LDAP hierarchy could have one or many levels. You might decide to set up a single-level hierarchy in which Employee1 is considered a member of Example Corporation. Or you could set up a multi-level hierarchy in which Employee1 is considered to be a member of the department Engineering, which is a member of an organizational unit called People, which is itself a member of Example Corporation. See the following figure for an example of a multi-level hierarchy.

A multi-level hierarchy has more detail, but searches return results more quickly in a single-level hierarchy.

Figure 54: A Multi-Level LDAP Hierarchy



Search the LDAP Hierarchy

The ASA lets you tailor the search within the LDAP hierarchy. You configure the following three fields on the ASA to define where in the LDAP hierarchy that your search begins, the extent, and the type of information you are looking for. Together, these fields limit the search of the hierarchy to only the part that includes the user permissions.

- LDAP Base DN defines where in the LDAP hierarchy that the server should begin searching for user information when it receives an authorization request from the ASA.

- Search Scope defines the extent of the search in the LDAP hierarchy. The search proceeds this many levels in the hierarchy below the LDAP Base DN. You can choose to have the server search only the level immediately below it, or it can search the entire subtree. A single level search is quicker, but a subtree search is more extensive.
- Naming Attribute(s) defines the RDN that uniquely identifies an entry in the LDAP server. Common naming attributes can include cn (Common Name), sAMAccountName, and userPrincipalName.

The figure shows a sample LDAP hierarchy for Example Corporation. Given this hierarchy, you could define your search in different ways. The following table shows two sample search configurations.

In the first example configuration, when Employee1 establishes the IPsec tunnel with LDAP authorization required, the ASA sends a search request to the LDAP server, indicating it should search for Employee1 in the Engineering group. This search is quick.

In the second example configuration, the ASA sends a search request indicating that the server should search for Employee1 within Example Corporation. This search takes longer.

Table 40: Example Search Configurations

No.	LDAP Base DN	Search Scope	Naming Attribute	Result
1	group= Engineering,ou=People,dc=ExampleCorporation, dc=com	One Level	cn=Employee1	Quicker search
2	dc=ExampleCorporation,dc=com	Subtree	cn=Employee1	Longer search

Bind to an LDAP Server

The ASA uses the login DN and login password to establish trust (bind) with an LDAP server. When performing a Microsoft Active Directory read-only operation (such as authentication, authorization, or group search), the ASA can bind using a login DN with fewer privileges. For example, the login DN can be a user whose AD “Member Of” designation is part of Domain Users. For VPN password management operations, the login DN needs elevated privileges, and must be part of the Account Operators AD group.

The following is an example of a login DN:

```
cn=Binduser1,ou=Admins,ou=Users,dc=company_A,dc=com
```

The ASA supports the following authentication methods:

- Simple LDAP authentication with an unencrypted password on port 389
- Secure LDAP (LDAP-S) on port 636
- Simple Authentication and Security Layer (SASL) MD5
- SASL Kerberos

The ASA does not support anonymous authentication.

Note As an LDAP client, the ASA does not support the transmission of anonymous binds or requests.

LDAP Attribute Maps

The ASA can use an LDAP directory for authenticating users for:

- VPN remote access users
- Firewall network access/cut-through-proxy sessions
- Setting policy permissions (also called authorization attributes), such as ACLs, bookmark lists, DNS or WINS settings, and session timers.
- Setting the key attributes in a local group policy

The ASA uses LDAP attribute maps to translate native LDAP user attributes to Cisco ASA attributes. You can bind these attribute maps to LDAP servers or remove them. You can also show or clear attribute maps.

The LDAP attribute map does not support multi-valued attributes. For example, if a user is a member of several AD groups, and the LDAP attribute map matches more than one group, the value chosen is based on the alphabetization of the matched entries.

To use the attribute mapping features correctly, you need to understand LDAP attribute names and values, as well as the user-defined attribute names and values.

The names of frequently mapped LDAP attributes and the type of user-defined attributes that they would commonly be mapped to include the following:

- IETF-Radius-Class (Group_Policy in ASA version 8.2 and later)—Sets the group policy based on the directory department or user group (for example, Microsoft Active Directory memberOf) attribute value. The group policy attribute replaced the IETF-Radius-Class attribute with ASDM version 6.2/ASA version 8.2 or later.
- IETF-Radius-Filter-Id—Applies an access control list or ACL to VPN clients, IPsec, and SSL.
- IETF-Radius-Framed-IP-Address—Assigns a static IP address assigned to a VPN remote access client, IPsec, and SSL.
- Banner1—Displays a text banner when the VPN remote access user logs in.
- Tunneling-Protocols—Allows or denies the VPN remote access session based on the access type.

  Note A single LDAP attribute map may contain one or many attributes. You can only map one LDAP attribute from a specific LDAP server.

Guidelines for LDAP Servers for AAA

This section includes the guidelines and limitations that you should check before configuring LDAP servers for AAA.

IPv6

The AAA server must use an IPv4 address, but endpoints can use IPv6.

Additional Guidelines

- The DN configured on the ASA to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACL on the default password policy.
- You must configure LDAP over SSL to enable password management with Microsoft Active Directory and Sun servers.
- The ASA does not support password management with Novell, OpenLDAP, and other LDAPv3 directory servers.
- Beginning with Version 7.1(x), the ASA performs authentication and authorization using the native LDAP schema, and the Cisco schema is no longer needed.
- You can have up to 100 LDAP server groups in single mode or 4 LDAP server groups per context in multiple mode.
- Each group can have up to 16 LDAP servers in single mode or 4 LDAP servers in multiple mode.
- When a user logs in, the LDAP servers are accessed one at a time, starting with the first server that you specify in the configuration, until a server responds. If all servers in the group are unavailable, the ASA tries the local database if you configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the ASA continues to try the LDAP servers.

Configure LDAP Servers for AAA

This section describes how to configure LDAP servers for AAA.

Procedure

Step 1 Configure LDAP attribute maps. See Configure LDAP Attribute Maps, on page 817.

Step 2 Add an LDAP server group. See Configure LDAP Server Groups, on page 819.

Step 3 (Optional) Configure authorization from an LDAP server that is separate and distinct from the authentication mechanism. See Configure Authorization with LDAP for VPN, on page 822.

Configure LDAP Attribute Maps

To configure LDAP attribute maps, perform the following steps:

Procedure

Step 1 Create an unpopulated LDAP attribute map table.

ldap-attribute-map *map-name*

Example:

```
ciscoasa(config)# ldap-attribute-map att_map_1
```

Step 2 Map the user-defined attribute name department to the Cisco attribute.

map-name *user-attribute-name Cisco-attribute-name*

Example:

```
ciscoasa(config-ldap-attribute-map)# map-name department IETF-Radius-Class
```

Step 3 Map the user-defined map value department to the user-defined attribute value and the Cisco attribute value.

map-value *user-attribute-name Cisco-attribute-name*

Example:

```
ciscoasa(config-ldap-attribute-map)# map-value department Engineering group1
```

Step 4 Identify the server and the AAA server group to which it belongs.

aaa-server *server_group* [*interface_name*] **host** *server_ip*

Example:

```
ciscoasa(config)# aaa-server ldap_dir_1 host 10.1.1.4
```

Step 5 Bind the attribute map to the LDAP server.

ldap-attribute-map *map-name*

Example:

```
ciscoasa(config-aaa-server-host)# ldap-attribute-map att_map_1
```

Examples

The following example shows how to limit management sessions to the ASA based on an LDAP attribute called accessType. The accessType attribute may have one of these values:

- VPN
- admin
- helpdesk

The following example shows how each value is mapped to one of the valid IETF-Radius-Service-Type attributes that the ASA supports: remote-access (Service-Type 5) Outbound, admin (Service-Type 6) Administrative, and nas-prompt (Service-Type 7) NAS Prompt.

```
ciscoasa(config)# ldap attribute-map MGMT
ciscoasa(config-ldap-attribute-map)# map-name accessType IETF-Radius-Service-Type
ciscoasa(config-ldap-attribute-map)# map-value accessType VPN 5
ciscoasa(config-ldap-attribute-map)# map-value accessType admin 6
ciscoasa(config-ldap-attribute-map)# map-value accessType helpdesk 7

ciscoasa(config-ldap-attribute-map)# aaa-server LDAP protocol ldap
ciscoasa(config-aaa-server-group)# aaa-server LDAP (inside) host 10.1.254.91
ciscoasa(config-aaa-server-host)# ldap-base-dn CN=Users,DC=cisco,DC=local
ciscoasa(config-aaa-server-host)# ldap-scope subtree
ciscoasa(config-aaa-server-host)# ldap-login-password test
ciscoasa(config-aaa-server-host)# ldap-login-dn CN=Administrator,CN=Users,DC=cisco,DC=local
ciscoasa(config-aaa-server-host)# server-type auto-detect
ciscoasa(config-aaa-server-host)# ldap-attribute-map MGMT
```

The following example shows how to display the complete list of Cisco LDAP attribute names:

```
ciscoasa(config)# ldap attribute-map att_map_1
ciscoasa(config-ldap-attribute-map)# map-name att_map_1?

ldap mode commands/options:
cisco-attribute-names:
  Access-Hours
  Allow-Network-Extension-Mode
  Auth-Service-Type
  Authenticated-User-Idle-Timeout
  Authorization-Required
  Authorization-Type
 :
 :
  X509-Cert-Data
ciscoasa(config-ldap-attribute-map)#
```

Configure LDAP Server Groups

To create and configure an LDAP server group, then add an LDAP server to that group, perform the following steps:

Before you begin

You must add an attribute map before you may add an LDAP server to an LDAP server group.

Procedure

Step 1 Identify the server group name and the protocol.

aaa-server *server_tag* **protocol ldap**

Example:

```
ciscoasa(config)# aaa-server servergroup1 protocol ldap
ciscoasa(config-aaa-server-group)#
```

When you enter the **aaa-server protocol** command, you enter aaa-server group configuration mode.

Step 2 Specify the maximum number of failed AAA transactions with an LDAP server in the group before trying the next server.

max-failed-attempts *number*

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

The *number* argument can range from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only) to configure the fallback mechanism, and all the servers in the group fail to respond, or their responses are invalid, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

Step 3 Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode deadtime 20
```

The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive.

The **deadtime** *minutes* keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

Step 4 Identify the LDAP server and AAA server group to which it belongs.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config)# aaa-server servergroup1 outside host 10.10.1.1
```

If you do not specify an **(***interface_name***)**, then the ASA uses the **inside** interface by default.

When you enter the **aaa-server host** command, you enter aaa-server host configuration mode. As needed, use host configuration mode commands to further configure the AAA server.

The following table lists the available commands for LDAP servers, and whether or not a new LDAP server definition has a default value for that command. If no default value is provided (indicated by “—”), use the command to specify the value.

Table 41: Host Mode Commands and Defaults

Command	Default Value	Description
ldap-attribute-map	—	—
ldap-base-dn	—	—
ldap-login-dn	—	—
ldap-login-password	—	—
ldap-naming-attribute	—	—
ldap-over-ssl	636	If not set, the ASA uses sAMAccountName for LDAP requests. Whether using SASL or plain text, you can secure communications between the ASA and the LDAP server with SSL. If you do not configure SASL, we strongly recommend that you secure LDAP communications with SSL.
ldap-scope	—	—
sasl-mechanism	—	—
server-port	389	—
server-type	autodiscovery	If autodetection fails to determine the LDAP server type, and you know the server is either a Microsoft, Sun or generic LDAP server, you can manually configure the server type.
timeout	10 seconds	—

Example

The following example shows how to configure an LDAP server group named watchdogs and add an LDAP server to the group. Because the example does not define a retry interval or the port that the LDAP server listens to, the ASA uses the default values for these two server-specific parameters.

```
ciscoasa(config)# aaa-server watchdogs protocol ldap
ciscoasa(config-aaa-server-group)# aaa-server watchdogs host 192.168.3.4
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)#
```

Configure Authorization with LDAP for VPN

When LDAP user authentication for VPN access has succeeded, the ASA queries the LDAP server, which returns LDAP attributes. These attributes generally include authorization data that applies to the VPN session. Using LDAP in this way accomplishes authentication and authorization in a single step.

There may be cases, however, where you require authorization from an LDAP directory server that is separate and distinct from the authentication mechanism. For example, if you use an SDI or certificate server for authentication, no authorization information is returned. For user authorizations in this case, you can query an LDAP directory after successful authentication, accomplishing authentication and authorization in two steps.

To set up VPN user authorization using LDAP, perform the following steps.

Procedure

Step 1 Create an IPsec remote access tunnel group named remotegrp.

tunnel-group *groupname*

Example:

```
ciscoasa(config)# tunnel-group remotegrp
```

Step 2 Associate the server group and the tunnel group.

tunnel-group *groupname* **general-attributes**

Example:

```
ciscoasa(config)# tunnel-group remotegrp general-attributes
```

Step 3 Assign a new tunnel group to a previously created AAA server group for authorization.

authorization-server-group *group-tag*

Example:

```
ciscoasa(config-general)# authorization-server-group ldap_dir_1
```

Example

While there are other authorization-related commands and options available for specific requirements, the following example shows commands for enabling user authorization with LDAP. The example then creates an IPsec remote access tunnel group named remote-1, and assigns that new tunnel group to the previously created ldap_dir_1 AAA server group for authorization:

```
ciscoasa(config)# tunnel-group remote-1 type ipsec-ra
ciscoasa(config)# tunnel-group remote-1 general-attributes
ciscoasa(config-general)# authorization-server-group ldap_dir_1
ciscoasa(config-general)#
```

After you complete this configuration work, you can then configure additional LDAP authorization parameters such as a directory password, a starting point for searching a directory, and the scope of a directory search by entering the following commands:

```
ciscoasa(config)# aaa-server ldap_dir_1 protocol ldap
ciscoasa(config-aaa-server-group)# aaa-server ldap_dir_1 host 10.1.1.4
ciscoasa(config-aaa-server-host)# ldap-login-dn obscurepassword
ciscoasa(config-aaa-server-host)# ldap-base-dn starthere
ciscoasa(config-aaa-server-host)# ldap-scope subtree
ciscoasa(config-aaa-server-host)#
```

Monitoring LDAP Servers for AAA

See the following commands for monitoring LDAP servers for AAA:

- **show aaa-server**

  This command shows the configured AAA server statistics. Use the **clear aaa-server statistics** command to clear the AAA server statistics.

- **show running-config aaa-server**

  This command shows the AAA server running configuration. Use the **clear configure aaa-server** command to clear AAA server configuration.

History for LDAP Servers for AAA

Table 42: History for AAA Servers

Feature Name	Platform Releases	Description
LDAP Servers for AAA	7.0(1)	LDAP Servers describe support for AAA and how to configure LDAP servers. We introduced the following commands: **username**, **aaa authorization exec authentication-server**, **aaa authentication console LOCAL**, **aaa authorization exec LOCAL**, **service-type, ldap attribute-map**, **aaa-server protocol**, **aaa authentication telnet | ssh | serial**} **console LOCAL**, **aaa authentication http console LOCAL**, **aaa authentication enable console LOCAL**, **max-failed-attempts**, **reactivation-mode**, **accounting-mode simultaneous**, **aaa-server host**, **authorization-server-group**, **tunnel-group**, **tunnel-group general-attributes**, **map-name**, **map-value**, **ldap-attribute-map**.

CHAPTER 34

Kerberos Servers for AAA

The following topics explain how to configure Kerberos servers used in AAA. You can use Kerberos servers for the authentication of management connections, network access, and VPN user access.

Guidelines for Kerberos Servers for AAA

- You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode.
- Each group can have up to 16 servers in single mode or 4 servers in multiple mode. When a user logs in, the servers are accessed one at a time starting with the first server you specify in the configuration, until a server responds.

Configure Kerberos Servers for AAA

The following topics explain how to configure Kerberos server groups. You can then use these groups when configuring management access or VPNs.

Configure Kerberos AAA Server Groups

If you want to use a Kerberos server for authentication, you must first create at least one Kerberos server group and add one or more servers to each group.

Procedure

Step 1 Create the Kerberos AAA server group and enter aaa-server-group configuration mode.

aaa-server *server_group_name* **protocol kerberos**

Example:

```
ciscoasa(config)# aaa-server watchdog protocol kerberos
```

Step 2 (Optional.) Specify the maximum number of failed AAA transactions with a AAA server in the group before trying the next server.

max-failed-attempts *number*

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

The *number* argument can range from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, or their responses are invalid, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

Step 3 (Optional.) Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode depletion deadtime 20
```

The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive. This is the default mode.

The **deadtime** *minutes* keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

Example

The following example creates a Kerberos server group named watchdogs, adds a server, and sets the realm to EXAMPLE.COM.

```
hostname(config)# aaa-server watchdogs protocol kerberos
hostname(config-aaa-server-group)# aaa-server watchdogs host 192.168.3.4
hostname(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
hostname(config-aaa-server-host)# exit
hostname(config)#
```

Add Kerberos Servers to a Kerberos Server Group

Before you can use a Kerberos server group, you must add at least one Kerberos server to the group.

Procedure

Step 1 Add the Kerberos server to the Kerberos server group.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config-aaa-server-group)# aaa-server servergroup1 outside host 10.10.1.1
```

If you do not specify an interface, then the ASA uses the **inside** interface by default.

Step 2 Specify the timeout value for connection attempts to the server.

timeout *seconds*

Specify the timeout interval (1-300 seconds) for the server; the default is 10 seconds. For each AAA transaction the ASA retries connection attempts (based on the interval defined on the **retry-interval** command) until the timeout is reached. If the number of consecutive failed transactions reaches the limit specified on the max-failed-attempts command in the AAA server group, the AAA server is deactivated and the ASA starts sending requests to another AAA server if it is configured.

Example:

```
ciscoasa(config-aaa-server-host)# timeout 15
```

Step 3 Specify the retry interval, which is the time the system waits before retrying a connection request.

retry-interval *seconds*

You can specify 1-10 seconds. The default is 10.

Example:

```
ciscoasa(config-aaa-server-host)# retry-interval 6
```

Step 4 Specify the server port if it is different from the default Kerberos port, which is TCP/88. The ASA contacts the Kerberos server on this port.

server-port *port_number*

Example:

```
ciscoasa(config-aaa-server-host)# server-port 8888
```

Step 5 Configure the Kerberos realm.

kerberos-realm *name*

Kerberos realm names use numbers and upper case letters only, and can be up to 64 characters. The name should match the output of the Microsoft Windows **set USERDNSDOMAIN** command when it is run on the Active Directory server for the Kerberos realm. In the following example, EXAMPLE.COM is the Kerberos realm name:

```
C:\>set USERDNSDOMAIN
USERDNSDOMAIN=EXAMPLE.COM
```

Although the ASA accepts lower case letters in the name, it does not translate lower case letters to upper case letters. Be sure to use upper case letters only.

Example:

```
ciscoasa(config-asa-server-group)# kerberos-realm EXAMPLE.COM
```

Example

```
hostname(config)# aaa-server watchdogs protocol kerberos
hostname(config-aaa-server-group)# aaa-server watchdogs host 192.168.3.4
ciscoasa(config-aaa-server-host)# timeout 9
ciscoasa(config-aaa-server-host)# retry 7
ciscoasa(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)#
```

Monitor Kerberos Servers for AAA

You can use the following commands to monitor and clear Kerberos-related information.

- **show aaa-server**

  Shows the AAA server statistics. Use the **clear aaa-server statistics** command to clear the server statistics.

- **show running-config aaa-server**

  Shows the AAA servers that are configured for the system. Use the **clear configure aaa-server** command to remove the AAA server configuration.

- **show aaa kerberos** [**username** *user*]

  Shows all Kerberos tickets, or tickets for a given username.

- **clear aaa kerberos tickets** [**username** *user*]

  Clears all Kerberos tickets, or tickets for a given username.

History for Kerberos Servers for AAA

Feature Name	Platform Releases	Description
Kerberos Servers	7.0(1)	Support for Kerberos servers for AAA. We introduced the following commands: **aaa-server protocol**, **max-failed-attempts**, **reactivation-mode**, **aaa-server host**, **kerberos-realm**, **server-port**, **clear aaa-server statistics**, **clear configure aaa-server**, **show aaa-server**, **show running-config aaa-server**, **timeout**.

CHAPTER 35

RSA SecurID Servers for AAA

The following topics explain how to configure RSA SecurID servers used in AAA. The RSA SecureID servers are also known as SDI servers, because SDI is the protocol used to communicate with them. You can use RSA SecurID servers for the authentication of management connections, network access, and VPN user access.

About RSA SecurID Servers

You can use RSA SecurID servers either directly for authentication, or indirectly, as a second factor for authentication. In the latter case, you would configure the relationship to the SecurID server between the SecurID server and your RADIUS server, and configure the ASA to use the RADIUS server.

But, if you want to directly authenticate against the SecurID server, you would create a AAA server group for the SDI protocol, which is the protocol used to communicate with these servers.

When you use SDI, you need only specify the primary SecurID server when you create the AAA server group. The ASA will retrieve the sdiconf.rec file, which lists all of the SecurID server replicas, when it first connects to the server. The ASA can then use these replicas for authentication if the primary server does not respond.

In addition, you must register the ASA as an authentication agent in the RSA Authentication Manager. Authentication attempts will fail until you register the ASA.

Guidelines for RSA SecurID Servers for AAA

- You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode.

- Each group can have up to 16 servers in single mode or 4 servers in multiple mode. When a user logs in, the servers are accessed one at a time starting with the first server you specify in the configuration, until a server responds.

Configure RSA SecurID Servers for AAA

The following topics explain how to configure RSA SecurID server groups. You can then use these groups when configuring management access or VPNs.

Configure RSA SecurID AAA Server Groups

If you want to use direct communication with an RSA SecurID server for authentication, you must first create at least one SDI server group and add one or more servers to each group. If you are using the SecurID server in a proxy relationship with a RADIUS server, you do not need to configure an SDI AAA server group on the ASA.

Procedure

Step 1 Create the SDI AAA server group and enter aaa-server-group configuration mode.

aaa-server *server_group_name* **protocol sdi**

Example:

```
ciscoasa(config)# aaa-server watchdog protocol sdi
```

Step 2 (Optional.) Specify the maximum number of failed AAA transactions with a AAA server in the group before trying the next server.

max-failed-attempts *number*

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

The *number* argument can range from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, or their responses are invalid, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

Step 3 (Optional.) Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode depletion deadtime 20
```

The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive. This is the default mode.

The **deadtime** *minutes* keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

Add RSA SecurID Servers to an SDI Server Group

Before you can use an SDI server group, you must add at least one RSA SecurID server to the group.

Servers in an SDI server group use the authentication and server management protocol (ACE) to communicate with the ASA.

Procedure

Step 1 Add the RSA SecurID server to the SDI server group.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config-aaa-server-group)# aaa-server servergroup1 outside host 10.10.1.1
```

If you do not specify an interface, then the ASA uses the **inside** interface by default.

Step 2 Specify the timeout value for connection attempts to the server.

timeout *seconds*

Specify the timeout interval (1-300 seconds) for the server; the default is 10 seconds. For each AAA transaction the ASA retries connection attempts (based on the interval defined on the **retry-interval** command) until the timeout is reached. If the number of consecutive failed transactions reaches the limit specified on the **max-failed-attempts** command in the AAA server group, the AAA server is deactivated and the ASA starts sending requests to another AAA server if it is configured.

Example:

```
ciscoasa(config-aaa-server-host)# timeout 15
```

Step 3 Specify the retry interval, which is the time the system waits before retrying a connection request.

retry-interval *seconds*

You can specify 1-10 seconds. The default is 10.

Example:

```
ciscoasa(config-aaa-server-host)# retry-interval 6
```

Step 4 Specify the server port if it is different from the default RSA SecurID port, which is TCP/5500. The ASA contacts the RSA SecurID server on this port.

server-port *port_number*

Example:

```
ciscoasa(config-aaa-server-host)# server-port 5555
```

Monitor RSA SecurID Servers for AAA

You can use the following commands to monitor and clear RSA SecurID-related information.

- **show aaa-server**

  Shows the AAA server statistics. Use the **clear aaa-server statistics** command to clear the server statistics.

- **show running-config aaa-server**

  Shows the AAA servers that are configured for the system. Use the **clear configure aaa-server** command to remove the AAA server configuration.

History for RSA SecurID Servers for AAA

Feature Name	Platform Releases	Description
SecurID Servers	7.2(1)	Support for SecurID servers for AAA for management authentication. SecurID was supported in previous releases for VPN authentication.

PART VII

System Administration

CHAPTER 36

Management Access

This chapter describes how to access the Cisco ASA for system management through Telnet, SSH, and HTTPS (using ASDM), how to authenticate and authorize users, and how to create login banners.

Configure Management Remote Access

This section describes how to configure ASA access for ASDM, Telnet, or SSH, and other management parameters such as a login banner.

Configure SSH Access

To identify the client IP addresses and define a user allowed to connect to the ASA using SSH, perform the following steps. See the following guidelines:

- To access the ASA interface for SSH access, you do not also need an access rule allowing the host IP address. You only need to configure SSH access according to this section.

- SSH access to an interface other than the one from which you entered the ASA is not supported. For example, if your SSH host is located on the outside interface, you can only initiate a management connection directly to the outside interface. The only exception to this rule is through a VPN connection. See Configure Management Access Over a VPN Tunnel, on page 846.

- The ASA allows a maximum of 5 concurrent SSH connections per context/single mode, with a maximum of 100 connections divided among all contexts. However, because configuration commands might obtain locks on resources being changed, you should make changes in one SSH session at a time to ensure all changes are applied correctly.

- (8.4 and later) The SSH default username is no longer supported. You can no longer connect to the ASA using SSH with the **pix** or **asa** username and the login password. To use SSH, you must configure AAA authentication using the **aaa authentication ssh console LOCAL** command; then define a local user by entering the **username** command. If you want to use a AAA server for authentication instead of the local database, we recommend also configuring local authentication as a backup method.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Generate an RSA key pair, which is required for SSH (for physical ASAs only).

crypto key generate rsa modulus*size*

- *size*—The size in bits is 512, 768, 1024, 2048, or 4096. We recommend a value of at least 2048. The larger the key size you specify, the longer it takes to generate a key pair.

For the ASAv, the key pairs are automatically created after deployment.

Example:

```
ciscoasa(config)# crypto key generate rsa modulus 4096
```

Step 2 Save the keys to persistent flash memory.

write memory

Example:

```
ciscoasa(config)# write memory
```

Step 3 Create a user in the local database that can be used for SSH access. You can alternatively use a AAA server for user access, but a local username is recommended.

username *name* **password** *password* **privilege** *level*

Example:

```
ciscoasa(config)# username admin password Far$cape1999 privilege 15
```

By default, the privilege level is 2; enter a level between 0 and 15, where 15 has all privileges. **Note:** Do not use the **username** command **nopassword** option to avoid having to create a username with a password; the **nopassword** option allows *any* password to be entered, not no password. To restrict a user to public key authentication even though you assigned a password to the user, then do not also enable AAA authentication for password use according to this procedure.

Step 4 (Optional) Allow public key authentication for a user instead of/as well as password authentication, and enter the public key on the ASA:

username *name* **attributes**

ssh authentication {**pkf** | **publickey** *key*}

Example:

```
ciscoasa(config)# username admin attributes
ciscoasa(config-username)# ssh authentication pkf
```

```
Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by xxx@xxx from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnFaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77II==
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
```

For a local **username**, you can enable public key authentication instead of/as well as password authentication. You can generate a public key/private key pair using any SSH key generation software (such as ssh keygen) that can generate ssh-rsa raw keys (with no certificates). Enter the public key on the ASA. The SSH client then uses the private key (and the passphrase you used to create the key pair) to connect to the ASA.

For a **pkf** key, you are prompted to paste in a PKF formatted key, up to 4096 bits. Use this format for keys that are too large to paste inline in Base64 format. For example, you can generate a 4096-bit key using ssh keygen, then convert it to PKF, and use the **pkf** keyword to be prompted for the key. **Note:** You can use the **pkf** option with failover, but the PKF key is not automatically replicated to the standby system. You must enter the **write standby** command to synchronize the PKF key.

For a **publickey** *key*, the key is a Base64-encoded public key. You can generate the key using any SSH key generation software (such as ssh keygen) that can generate ssh-rsa raw keys (with no certificates).

Step 5 (For password access) Enable local (or AAA server) authentication for SSH access:

aaa authentication ssh console {**LOCAL** | *server_group* [**LOCAL**]}

Example:

```
ciscoasa(config)# aaa authentication ssh console LOCAL
```

This command does not affect local public key authentication for usernames with the **ssh authentication** command. The ASA implicitly uses the local database for public key authentication. This command only affects usernames with passwords. If you want to allow either public key authentication or password use by a local user, then you need to explicitly configure local authentication with this command to allow password access.

Step 6 Identify the IP addresses from which the ASA accepts connections for each address or subnet, and the interface on which you can use SSH.

ssh *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface.

Unlike Telnet, you can SSH on the lowest security level interface.

Example:

```
ciscoasa(config)# ssh 192.168.3.0 255.255.255.0 inside
```

Step 7 (Optional) Set the duration for how long an SSH session can be idle before the ASA disconnects the session.

ssh timeout *minutes*

Example:

```
ciscoasa(config)# ssh timeout 30
```

Set the timeout from 1 to 60 minutes. The default is 5 minutes. The default duration is too short in most cases, and should be increased until all pre-production testing and troubleshooting have been completed.

Step 8 (Optional) Limit access to SSH version 1 or 2. By default, SSH allows both versions 1 and 2.

ssh version *version_number*

Example:

```
ciscoasa(config)# ssh version 2
```

Step 9 (Optional) Set the Diffie-Hellman (DH) key exchange mode:

ssh key-exchange group {**dh-group1-sha1** | **dh-group14-sha1**}

Example:

```
ciscoasa(config)# ssh key-exchange group dh-group14-sha1
```

The default is **dh-group1-sha1**

The DH key exchange provides a shared secret that cannot be determined by either party alone. The key exchange is combined with a signature and the host key to provide host authentication. This key-exchange method provides explicit server authentication. For more information about using DH key-exchange methods, see RFC 4253.

Examples

The following example shows how to authenticate using a PKF formatted key:

```
ciscoasa(config)# crypto key generate rsa modulus 4096
ciscoasa(config)# write memory
ciscoasa(config)# username exampleuser1 password examplepassword1 privilege 15
ciscoasa(config)# username exampleuser1 attributes
ciscoasa(config-username)# ssh authentication pkf
Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by xxx@xxx from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnFaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
```

```
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77II==
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
ciscoasa(config)#
```

The following example generates a shared key for SSH on a Linux or Macintosh system, and imports it to the ASA:

1. Generate the RSA public and private keys for 4096 bits on your computer:

```
jcrichton-mac:~ john$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/john/.ssh/id_rsa):
/Users/john/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): pa$$phrase
Enter same passphrase again: pa$$phrase
Your identification has been saved in /Users/john/.ssh/id_rsa.
Your public key has been saved in /Users/john/.ssh/id_rsa.pub.
The key fingerprint is:
c0:0a:a2:3c:99:fc:00:62:f1:ee:fa:f8:ef:70:c1:f9 john@jcrichton-mac
The key's randomart image is:
+--[ RSA 4096]----+
| .               |
|  o  .           |
|+...  o          |
|B.+.....         |
|.B ..+  S        |
|  =   o          |
|   + . E         |
|  o o            |
| ooooo           |
+-----------------+
```

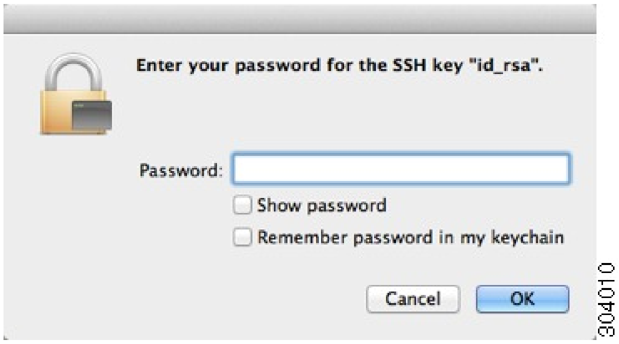
2. Convert the key to PKF format:

```
jcrichton-mac:~ john$ cd .ssh
jcrichton-mac:.ssh john$ ssh-keygen -e -f id_rsa.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by john@jcrichton-mac from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnDaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77IQ==
---- END SSH2 PUBLIC KEY ----
jcrichton-mac:.ssh john$
```

3. Copy the key to your clipboard.
4. In ASDM, choose **Configuration** > **Device Management** > **Users/AAA** > **User Accounts**, select the username and then click **Edit**. Click **Public Key Using PKF** and paste the key into the window:
5. Verify the user can SSH to the ASA. For the password, enter the SSH key password you specified when you created the key pair.

```
jcrichton-mac:.ssh john$ ssh test@10.86.118.5
The authenticity of host '10.86.118.5 (10.86.118.5)' can't be established.
RSA key fingerprint is 39:ca:ed:a8:75:5b:cc:8e:e2:1d:96:2b:93:b5:69:94.
Are you sure you want to continue connecting (yes/no)? yes
```

The following dialog box appears for you to enter your passphrase:



Enter your password for the SSH key "id_rsa".

Password:

□ Show password

□ Remember password in my keychain

Meanwhile, in the terminal session:

```
Warning: Permanently added '10.86.118.5' (RSA) to the list of known hosts.
Identity added: /Users/john/.ssh/id_rsa (/Users/john/.ssh/id_rsa)
Type help or '?' for a list of available commands.
asa>
```

Configure Telnet Access

To identify the client IP addresses allowed to connect to the ASA using Telnet, perform the following steps. See the following guidelines:

- To access the ASA interface for Telnet access, you do not also need an access rule allowing the host IP address. You only need to configure Telnet access according to this section.
- Telnet access to an interface other than the one from which you entered the ASA is not supported. For example, if your Telnet host is located on the outside interface, you can only initiate a Telnet connection directly to the outside interface. The only exception to this rule is through a VPN connection. See Configure Management Access Over a VPN Tunnel, on page 846.
- You cannot use Telnet to the lowest security interface unless you use Telnet inside a VPN tunnel.
- The ASA allows a maximum of 5 concurrent Telnet connections per context/single mode, with a maximum of 100 connections divided among all contexts.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.
- To gain access to the ASA CLI using Telnet, enter the login password set by the **password** command. You must manually set the password before using Telnet.

Procedure

Step 1 Identify the IP addresses from which the ASA accepts connections for each address or subnet on the specified interface.

telnet *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface.

If there is only one interface, you can configure Telnet to access that interface as long as the interface has a security level of 100.

Example:

```
ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside
```

Step 2 Set the duration for how long a Telnet session can be idle before the ASA disconnects the session.

telnet timeout *minutes*

Example:

```
ciscoasa(config)#  telnet timeout 30
```

Set the timeout from 1 to 1440 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting have been completed.

Examples

The following example shows how to let a host on the inside interface with an address of 192.168.1.2 access the ASA:

```
ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside
```

The following example shows how to allow all users on the 192.168.3.0 network to access the ASA on the inside interface:

```
ciscoasa(config)# telnet 192.168.3.0. 255.255.255.255 inside
```

Configure HTTPS Access for ASDM, Other Clients

To use ASDM or other HTTPS clients such as CSM, you need to enable the HTTPS server, and allow HTTPS connections to the ASA. HTTPS access is enabled as part of the factory default configuration. To configure HTTPS access, perform the following steps. See the following guidelines:

- To access the ASA interface for HTTPS access, you do not also need an access rule allowing the host IP address. You only need to configure HTTPS access according to this section. If, however, you configure HTTP redirect to redirect HTTP connections to HTTPS automatically, you must enable an access rule to allow HTTP; otherwise, the interface cannot listen to the HTTP port.
- Management access to an interface other than the one from which you entered the ASA is not supported. For example, if your management host is located on the outside interface, you can only initiate a management connection directly to the outside interface. The only exception to this rule is through a VPN connection. See Configure Management Access Over a VPN Tunnel, on page 846.
- In single context mode, you can have a maximum 30 ASDM concurrent sessions. In multiple context mode, you can have a maximum of 5 concurrent ASDM sessions per context, with a maximum of 32 ASDM instances among all contexts.

  ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the multiple-context mode system limit of 32 ASDM sessions represents a limit of 64 HTTPS sessions.
- The ASA allows a maximum of 6 concurrent non-ASDM HTTPS sessions in single context mode or per context, if available, with a maximum or 100 HTTPS sessions among all contexts.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Identify the IP addresses from which the ASA accepts HTTPS connections for each address or subnet on the specified interface.

http *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface.

Example:

```
ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside
```

Step 2 Enable the HTTPS server.

http server enable [*port*]

Example:

```
ciscoasa(config)# http server enable 444
```

By default, the port is 443. If you change the port number, be sure to include it in the ASDM access URL. For example, if you change the port number to 444, enter the following URL:

https://10.1.1.1:444

Step 3 (Optional) Set connection and session timeouts.

http server idle-timeout*minutes*

http server session-timeout*minutes*

- **http server idle-timeout** *minutes*—Set the idle timeout for ASDM connections, from 1-1440 minutes. The default is 20 minutes. The ASA disconnects an ASDM connection that is idle for the set period of time.
- **http server session-timeout** *minutes*—Set the session timeout for ASDM sessions, from 1-1440 minutes. This timeout is disabled by default. The ASA disconnects an ASDM session that exceeds the set period of time.

Example:

```
ciscoasa(config)# http server idle-timeout 30
ciscoasa(config)# http server session-timeout 120
```

Examples

The following example shows how to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access ASDM:

```
ciscoasa(config)# http server enable
ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside
```

The following example shows how to allow all users on the 192.168.3.0/24 network to access ASDM on the inside interface:

```
ciscoasa(config)# http 192.168.3.0 255.255.255.0 inside
```

Configure HTTP Redirect for ASDM Access or Clientless SSL VPN

You must use HTTPS to connect to the ASA using ASDM or clientless SSL VPN. For your convenience, you can redirect HTTP management connections to HTTPS. For example, by redirecting HTTP, you can enter either **http://10.1.8.4/admin/** or **https://10.1.8.4/admin/** and still arrive at the ASDM launch page at the HTTPS address.

This feature only supports IPv4 redirection.

Before you begin

Normally, you do not need an access rule allowing the host IP address. However, for HTTP redirect, you must enable an access rule to allow HTTP; otherwise, the interface cannot listen to the HTTP port.

Procedure

Enable HTTP redirect:

http redirect *interface_name* [*port*]

Example:

```
ciscoasa(config)# http redirect outside 88
```

The *port* identifies the port from which the interface redirects HTTP connections. The default is 80.

Configure Management Access Over a VPN Tunnel

If your VPN tunnel terminates on one interface, but you want to manage the ASA by accessing a different interface, you must identify that interface as a management-access interface. For example, if you enter the ASA from the outside interface, this feature lets you connect to the inside interface using ASDM, SSH, Telnet, or SNMP; or you can ping the inside interface when entering from the outside interface.

Note For secure SNMP polling over a site-to-site VPN, include the IP address of the outside interface in the crypto map access-list as part of the VPN configuration.

VPN access to an interface other than the one from which you entered the ASA is not supported. For example, if your VPN access is located on the outside interface, you can only initiate a connection directly to the outside interface. You should enable VPN on the directly-accessible interface of the ASA and use name resolution so that you don’t have to remember multiple addresses.

Management access is available via the following VPN tunnel types: IPsec clients, IPsec Site-to-Site, and the AnyConnect SSL VPN client.

Procedure

Specify the name of the management interface that you want to access when entering the ASA from another interface.

management-access *management_interface*

Bridge group interfaces are not supported.

Example:

```
ciscoasa(config)# management-access inside
```

Change the Console Timeout

The console timeout sets how long a connection can remain in privileged EXEC mode or configuration mode; when the timeout is reached, the session drops into user EXEC mode. By default, the session does not time out. This setting does not affect how long you can remain connected to the console port, which never times out.

Procedure

Specify the idle time in minutes (0 through 60) after which the privileged session ends.

console timeout *number*

Example:

```
ciscoasa(config)# console timeout 0
```

The default timeout is 0, which means the session does not time out.

Customize a CLI Prompt

The ability to add information to a prompt allows you to see at-a-glance which ASA you are logged into when you have multiple modules. During a failover, this feature is useful when both ASAs have the same hostname.

In multiple context mode, you can view the extended prompt when you log in to the system execution space or the admin context. Within a non-admin context, you only see the default prompt, which is the hostname and the context name.

By default, the prompt shows the hostname of the ASA. In multiple context mode, the prompt also displays the context name. You can display the following items in the CLI prompt:

cluster-unit	Displays the cluster unit name. Each unit in a cluster can have a unique name.
context	(Multiple mode only) Displays the name of the current context.
domain	Displays the domain name.
hostname	Displays the hostname.
priority	Displays the failover priority as pri (primary) or sec (secondary).

state	Displays the traffic-passing state or role of the unit. For failover, the following values are displayed for the state keyword: • **act**—Failover is enabled, and the unit is actively passing traffic. • **stby**— Failover is enabled, and the unit is not passing traffic and is in a standby, failed, or other non-active state. • **actNoFailover**—Failover is not enabled, and the unit is actively passing traffic. • **stbyNoFailover**—Failover is not enabled, and the unit is not passing traffic. This might happen when there is an interface failure above the threshold on the standby unit. For clustering, the values for control and data are shown.

Procedure

Customize the CLI prompt by entering the following command:

prompt {[**hostname**] [**context**] [**domain**] [**slot**] [**state**] [**priority**] [**cluster-unit**]}

Example:

```
ciscoasa(config)# prompt hostname context slot state priority
ciscoasa/admin/pri/act(config)#
```

The order in which you enter the keywords determines the order of the elements in the prompt, which are separated by a slash (/).

Configure a Login Banner

You can configure a message to display when a user connects to the ASA, before a user logs in, or before a user enters privileged EXEC mode.

Before you begin

- From a security perspective, it is important that your banner discourage unauthorized access. Do not use the words “welcome” or “please,” as they appear to invite intruders in. The following banner sets the correct tone for unauthorized access:

  ```
  You have logged in to a secure device.
  If you are not authorized to access this device,
  log out immediately or risk possible criminal consequences.
  ```

- After a banner has been added, Telnet or SSH sessions to the ASA may close if:
    - There is not enough system memory available to process the banner message(s).
    - A TCP write error occurs when trying to display banner message(s).
- See RFC 2196 for guidelines about banner messages.

Procedure

Add a banner to display at one of three times: when a user first connects (message-of-the-day (motd)), when a user logs in (login), and when a user accesses privileged EXEC mode (exec).

banner {**exec** | **login** | **motd**} *text*

Example:

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).
```

When a user connects to the ASA, the message-of-the-day banner appears first, followed by the login banner and prompts. After the user successfully logs in to the ASA, the exec banner appears.

To add more than one line, precede each line by the **banner** command.

For the banner text:

- Spaces are allowed, but tabs cannot be entered using the CLI.
- There are no limits for banner length other than those for RAM and flash memory.
- You can dynamically add the hostname or domain name of the ASA by including the strings **$(hostname)** and **$(domain)**.
- If you configure a banner in the system configuration, you can use that banner text within a context by using the **$(system)** string in the context configuration.

Examples

The following examples show how to add a message-of-the-day banner:

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).
```

```
ciscoasa(config)# banner motd Contact me at admin@example.com for any issues.
```

Set a Management Session Quota

You can establish a maximum number of simultaneous ASDM, SSH, and Telnet sessions that are allowed on the ASA. If the maximum is reached, no additional sessions are allowed and a syslog message is generated. To prevent a system lockout, the management session quota mechanism cannot block a console session.

Before you begin

In multiple context mode, complete this procedure in the system execution space. To change from the context to the system configuration, enter the **changeto system**command.

Procedure

Step 1 Enter the following command:

quota management-session *number*

- *number*—Sets the aggregate number of sessions between 0 (unlimited) and 10000.

Example:

Example:

```
ciscoasa(config)# quota management-session 1000
```

Step 2 View the current sessions in use.

show quota management-session

Example:

```
ciscoasa(config)# show quota management-session

quota management-session limit 3
quota management-session warning level 2
quota management-session level 0
quota management-session high water 2
quota management-session errors 0
quota management-session warnings 0
```

Configure AAA for System Administrators

This section describes how to configure authentication, management authorization, and command authorization for system administrators.

Configure Management Authentication

Configure authentication for CLI and ASDM access.

About Management Authentication

How you log into the ASA depends on whether or not you enable authentication.

About SSH Authentication

See the following behavior for SSH access with and without authentication:

- No Authentication—SSH is not available without authentication.
- Authentication—When you enable SSH authentication, you enter the username and password as defined on the AAA server or local user database. For public key authentication, the ASA only supports the local database. If you configure SSH public key authentication, then the ASA uses the local database implicitly. You only need to explicitly configure SSH authentication when you use a username and password to log in. You access user EXEC mode.

About Telnet Authentication

See the following behavior for Telnet access with and without authentication:

- No Authentication—If you do not enable any authentication for Telnet, you do not enter a username; you enter the login password (set with the **password** command). There is no default password, so you must set one before you can Telnet to the ASA. You access user EXEC mode.
- Authentication—If you enable Telnet authentication, you enter the username and password as defined on the AAA server or local user database. You access user EXEC mode.

About ASDM Authentication

See the following behavior for ASDM access with and without authentication. You can also configure certificate authentication, with or without AAA authentication.

- No Authentication—By default, you can log into ASDM with a blank username and the enable password set by the **enable password** command, which is blank by default. We suggest that you change the enable password as soon as possible so that it does not remain blank; see Set the Hostname, Domain Name, and the Enable and Telnet Passwords, on page 503. Note that if you enter a username and password at the login screen (instead of leaving the username blank), ASDM checks the local database for a match.
- Certificate Authentication—(Single, routed mode only) You can require that the user have a valid certificate. Enter the certificate username and password, and the ASA validates the certificate against the PKI trustpoint.
- AAA Authentication—When you enable ASDM (HTTPS) authentication, you enter the username and password as defined on the AAA server or local user database. You can no longer use ASDM with a blank username and the enable password.
- AAA Authentication plus Certificate Authentication—(Single, routed mode only) When you enable ASDM (HTTPS) authentication, you enter the username and password as defined on the AAA server or local user database. If the username and password are different for the certificate authentication, you are prompted to enter them as well. You can opt to pre-fill the username derived from your certificate.

About Serial Authentication

See the following behavior for access to the serial console port with and without authentication:

- No Authentication—If you do not enable any authentication for serial access, you do not enter a username or password. You access user EXEC mode.
- Authentication—If you enable authentication for serial access, you enter the username and password as defined on the AAA server or local user database. You access user EXEC mode.

About Enable Authentication

To enter privileged EXEC mode after logging in, enter the **enable** command. How this command works depends on whether or not you enable authentication:

- No Authentication—If you do not configure enable authentication, enter the system enable password when you enter the **enable** command (set by the **enable password** command), which is blank by default. However, if you do not use enable authentication, after you enter the **enable** command, you are no longer logged in as a particular user, which can affect user-based features such as command authorization. To maintain your username, use enable authentication.
- Authentication—If you configure enable authentication, the ASA prompts you for your username and password as defined on the AAA server or local user database. This feature is particularly useful when you perform command authorization, in which usernames are important in determining the commands that a user can enter.

For enable authentication using the local database, you can use the **login** command instead of the **enable** command. The **login** command maintains the username, but requires no configuration to turn on authentication.

⚠

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged EXEC mode, you should configure command authorization. Without command authorization, users can access privileged EXEC mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can discourage the login command by using a AAA server for authentication instead of the local database, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged EXEC mode.

Sessions from the Host Operating System to the ASA

Some platforms support running the ASA as a separate application: for example, the ASASM on the Catalyst 6500, or the ASA on the Firepower 4100/9300. For sessions from the host operating system to the ASA, you can configure serial and Telnet authentication, depending on the type of connection.

In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet or serial authentication in the admin context, then authentication also applies to these sessions. The admin context AAA server or local user database is used in this instance.

Configure Authentication for CLI and ASDM Access

Before you begin

- Configure Telnet, SSH, or HTTP access.
- For external authentication, configure a AAA server group. For local authentication, add users to the local database.
- HTTP management authentication does not support the SDI protocol for a AAA server group.
- This feature does not affect SSH public key authentication for local usernames with the **ssh authentication** command. The ASA implicitly uses the local database for public key authentication. This feature only affects usernames with passwords. If you want to allow either public key authentication or password use by a local user, then you need to explicitly configure local authentication with this procedure to allow password access.

Procedure

Authenticate users for management access.

aaa authentication {**telnet** | **ssh** | **http** | **serial**} **console** {**LOCAL** | *server_group* [**LOCAL**]}

Example:

```
ciscoasa(config)# aaa authentication ssh console radius_1 LOCAL
ciscoasa(config)# aaa authentication http console radius_1 LOCAL
ciscoasa(config)# aaa authentication serial console LOCAL
```

The **telnet** keyword controls Telnet access. For the ASASM, this keyword also affects the session from the switch using the **session** command. The **ssh** keyword controls SSH access (password only; public key authentication implicitly uses the local database). The **http** keyword controls ASDM access. The **serial** keyword controls console port access. For the ASASM, for example, this keyword affects the virtual console accessed from the switch using the **service-module session** command.

If you use a AAA server group for authentication, you can configure the ASA to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (which is case sensitive). We recommend that you use the same username and password in the local database as the AAA server, because the ASA prompt does not give any indication which method is being used. You can alternatively use the local database as your primary method of authentication (with no fallback) by entering **LOCAL** alone.

Configure Enable Authentication (Privileged EXEC Mode)

You can authentication users when they enter the **enable** command.

Before you begin

See About Enable Authentication, on page 852.

Procedure

Choose one of the following options for authenticating users:

- To authenticate users with a AAA server or the local database, enter the following command:

  aaa authentication enable console {**LOCAL** | *server_group* [**LOCAL**]}

  Example:

  ```
  ciscoasa(config)# aaa authentication enable console LOCAL
  ```

  The user is prompted for the username and password.

  If you use a AAA server group for authentication, you can configure the ASA to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (which is case sensitive). We recommend that you use the same username and password in the local database as the AAA server, because the ASA prompt does not give any indication of which method is being used.

You can alternatively use the local database as your primary method of authentication (with no fallback) by entering **LOCAL** alone.

- To log in as a user from the local database, enter the following command:

  login

  Example:

  ```
  ciscoasa# login
  ```

  The ASA prompts for your username and password. After you enter your password, the ASA places you in the privilege level that the local database specifies.

  Users can log in with their own username and password to access privileged EXEC mode, so you do not have to provide the system enable password to everyone. To allow users to access privileged EXEC mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower.

Configure ASDM Certificate Authentication

You can require certificate authentication, with or without AAA authentication. The ASA validates the certificate against the PKI trustpoint.

Before you begin

This feature is supported in single, routed mode only.

Procedure

Step 1 Enable certificate authentication:

http authentication-certificate *interface_name*

Example:

```
ciscoasa(config)# http authentication-certificate outside
```

You configure certificate authentication for each interface, so that connections on a trusted/inside interface do not have to provide a certificate. You can use the command multiple times to enable certificate authentication on multiple interfaces.

Step 2 (Optional) Set the attribute used by ASDM to derive the username from the certificate:

http username-from-certificate{*primary-attr* [*secondary-attr*] | **use-entire-name** | **use-script**} [**pre-fill-username**]

Example:

```
ciscoasa(config)# http username-from-certificate CN pre-fill-username
```

By default, ASDM uses CN OU attributes.

- The *primary-attr* argument specifies the attribute to be used to derive the username. The *secondary-attr* argument specifies an additional attribute to use with the primary attribute to derive the username. You can use the following attributes:
    - C—Country
    - CN—Common Name
    - DNQ—DN qualifier
    - EA—Email Address
    - GENQ—Generational qualifier
    - GN—Given Name
    - I—Initials
    - L—Locality
    - N—Name
    - O—Organization
    - OU—Organizational Unit
    - SER—Serial Number
    - SN—Surname
    - SP—State/Province
    - T—Title
    - UID User ID
    - UPN—User Principal Name
- The **use-entire-name** keyword uses the entire DN name.
- The **use-script** keyword uses a Lua script generated by ASDM.
- The **pre-fill-username** keyword pre-fills the username when prompted for authentication. If the username is different from the one you initially typed in, a new dialog box appears with the username pre-filled. You can then enter the password for authentication.

Control CLI and ASDM Access with Management Authorization

The ASA lets you distinguish between administrative and remote-access users when they authenticate. User role differentiation can prevent remote access VPN and network access users from establishing an administrative connection to the ASA.

Before you begin

RADIUS or LDAP (mapped) users

When users are authenticated through LDAP, the native LDAP attributes and their values can be mapped to Cisco ASA attributes to provide specific authorization features. Configure Cisco VSA CVPN3000-Privilege-Level with a value between 0 and 15. and then map the LDAP attributes to Cisco VAS CVPN3000-Privilege-Level using the **ldap map-attributes** command.

The RADIUS IETF **service-type** attribute, when sent in an access-accept message as the result of a RADIUS authentication and authorization request, is used to designate which type of service is granted to the authenticated user

The RADIUS Cisco VSA **privilege-level** attribute (Vendor ID 3076, sub-ID 220), when sent in an access-accept message, is used to designate the level of privilege for the user.

TACACS+ users

Authorization is requested with “service=shell,” and the server responds with PASS or FAIL.

Local users

Set the **service-type** command for a given username. By default, the service-type is admin, which allows full access to any services specified by the **aaa authentication console**command.

Management Authorization Attributes

See the following table for AAA server types and valid values for management authorization. The ASA uses these values to determine the level of management access.

Management Level	RADIUS/LDAP (Mapped) Attributes	TACACS+ Attributes	Local Database Attributes
Full Access—Allows full access to any services specified by the **aaa authentication console** commands	Service-Type 6 (Administrative), Privilege-Level 1	PASS, privilege level 1	admin
Partial Access—Allows access to the CLI or ASDM when you configure the **aaa authentication console** commands. However, if you configure **enable** authentication with the **aaa authentication enable console** command, then the CLI user cannot access privileged EXEC mode using the **enable** command.	Service-Type 7 (NAS prompt), Privilege-Level 2 and higher The Framed (2) and Login (1) service types are treated the same way.	PASS, privilege level 2 and higher	nas-prompt
No Access—Denies management access. The user cannot use any services specified by the **aaa authentication console** commands(excluding the **serial** keyword; serial access is allowed). Remote access (IPsec and SSL) users can still authenticate and terminate their remote access sessions. All other service types (Voice, FAX, and so on) are treated the same way.	Service-Type 5 (Outbound)	FAIL	remote-access

Additional Guidelines

- Serial console access is not included in management authorization.
- You must also configure AAA authentication for management access to use this feature. See Configure Authentication for CLI and ASDM Access, on page 852.
- If you use external authentication, you must pre-configure a AAA server group before you enable this feature.
- HTTP authorization is supported in single, routed mode only.

Procedure

Step 1 Enable management authorization for Telnet and SSH:

aaa authorization exec {**authentication-server** | **LOCAL**} [**auto-enable**]

The **auto-enable** keyword lets administrators who have sufficient authorization privileges enter privileged EXEC mode automatically when they log in.

Example:

```
ciscoasa(config)# aaa authentication ssh console RADIUS
ciscoasa(config)# aaa authorization exec authentication-server auto-enable
```

Step 2 Enable management authorization for HTTPS (ASDM):

aaa authorization http console {**authentication-server** | **LOCAL**}

Example:

```
ciscoasa(config)# aaa authentication http console RADIUS
ciscoasa(config)# aaa authorization http console authentication-server
```

Step 3

Examples

The following example shows how to define an LDAP attribute map. In this example, the security policy specifies that users being authenticated through LDAP map the user record fields or parameters title and company to the IETF-RADIUS service-type and privilege-level, respectively.

```
ciscoasa(config)# ldap attribute-map admin-control
ciscoasa(config-ldap-attribute-map)# map-name title IETF-RADIUS-Service-Type
ciscoasa(config-ldap-attribute-map)# map-name company
```

The following example applies an LDAP attribute map to an LDAP AAA server:

```
ciscoasa(config)# aaa-server ldap-server (dmz1) host 10.20.30.1
ciscoasa(config-aaa-server-host)# ldap attribute-map admin-control
```

Configure Command Authorization

If you want to control access to commands, the ASA lets you configure command authorization, where you can determine which commands that are available to a user. By default when you log in, you can access user EXEC mode, which offers only minimal commands. When you enter the **enable** command (or the **login** command when you use the local database), you can access privileged EXEC mode and advanced commands, including configuration commands.

You can use one of two command authorization methods:

- Local privilege levels
- TACACS+ server privilege levels

About Command Authorization

You can enable command authorization so only authorized users can enter commands.

Supported Command Authorization Methods

You can use one of two command authorization methods:

- Local privilege levels—Configure the command privilege levels on the ASA. When a local, RADIUS, or LDAP (if you map LDAP attributes to RADIUS attributes) user authenticates for CLI access, the ASA places that user in the privilege level that is defined by the local database, RADIUS, or LDAP server. The user can access commands at the assigned privilege level and below. Note that all users access user EXEC mode when they first log in (commands at level 0 or 1). The user needs to authenticate again with the **enable** command to access privileged EXEC mode (commands at level 2 or higher), or they can log in with the **login** command (local database only).

  Note You can use local command authorization without any users in the local database and without CLI or **enable** authentication. Instead, when you enter the **enable** command, you enter the system enable password, and the ASA places you in level 15. You can then create enable passwords for every level, so that when you enter **enable** *n* (2 to 15), the ASA places you in level *n*. These levels are not used unless you enable local command authorization.

- TACACS+ server privilege levels—On the TACACS+ server, configure the commands that a user or group can use after authenticating for CLI access. Every command that a user enters at the CLI is validated with the TACACS+ server.

Security Contexts and Command Authorization

AAA settings are discrete per context, not shared among contexts.

When configuring command authorization, you must configure each security context separately. This configuration provides you the opportunity to enforce different command authorizations for different security contexts.

When switching between security contexts, administrators should be aware that the commands permitted for the username specified when they login may be different in the new context session or that command authorization may not be configured at all in the new context. Failure to understand that command authorizations may differ between security contexts could confuse an administrator.

Note The system execution space does not support AAA commands; therefore, command authorization is not available in the system execution space.

Command Privilege Levels

By default, the following commands are assigned to privilege level 0. All other commands are assigned to privilege level 15.

- **show checksum**
- **show curpriv**
- **enable**
- **help**
- **show history**
- **login**
- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

If you move any configure mode commands to a lower level than 15, be sure to move the **configure** command to that level as well, otherwise, the user cannot enter configuration mode.

Configure Local Command Authorization

Local command authorization lets you assign commands to one of 16 privilege levels (0 to 15). By default, each command is assigned either to privilege level 0 or 15. You can define each user to be at a specific privilege level, and each user can enter any command at the assigned privilege level or below. The ASA supports user privilege levels defined in the local database, a RADIUS server, or an LDAP server (if you map LDAP attributes to RADIUS attributes).

Procedure

Step 1 Assign a command to a privilege level.

privilege [**show** | **clear** | **cmd**] **level** *level* [**mode** {**enable** | **cmd**}] **command** *command*

Example:

```
ciscoasa(config)# privilege show level 5 command filter
```

Repeat this command for each command that you want to reassign.

The options in this command are the following:

- **show** | **clear** | **cmd**—These optional keywords let you set the privilege only for the show, clear, or configure form of the command. The configure form of the command is typically the form that causes a configuration change, either as the unmodified command (without the **show** or **clear** prefix) or as the **no** form. If you do not use one of these keywords, all forms of the command are affected.
- **level** *level*—A level between 0 and 15.
- **mode** {**enable** | **configure**}—If a command can be entered in user EXEC or privileged EXEC mode as well as configuration mode, and the command performs different actions in each mode, you can set the privilege level for these modes separately:
  - **enable**—Specifies both user EXEC mode and privileged EXEC mode.
  - **configure**—Specifies configuration mode, accessed using the **configure terminal** command.
- **command** *command*—The command you are configuring. You can only configure the privilege level of the *main* command. For example, you can configure the level of all **aaa** commands, but not the level of the **aaa authentication** command and the **aaa authorization** command separately.

Step 2 (Optional) Enable AAA users for command authorization. Without this command, the ASA only supports privilege levels for local database users and defaults all other types of users to level 15.

aaa authorization exec authentication-server [auto-enable]

Example:

```
ciscoasa(config)# aaa authorization exec authentication-server
```

This command also enables management authorization. See Control CLI and ASDM Access with Management Authorization, on page 855.

Step 3 Enable the use of local command privilege levels:

aaa authorization command LOCAL

Example:

```
ciscoasa(config)# aaa authorization command LOCAL
```

When you set command privilege levels, command authorization does not occur unless you configure command authorization with this command.

Examples

The **filter** command has the following forms:

- **filter** (represented by the **configure** option)

- **show running-config filter**
- **clear configure filter**

You can set the privilege level separately for each form, or set the same privilege level for all forms by omitting this option. The following example shows how to set each form separately:

```
ciscoasa(config)# privilege show level 5 command filter
ciscoasa(config)# privilege clear level 10 command filter
ciscoasa(config)# privilege cmd level 10 command filter
```

Alternatively, the following example shows how to set all filter commands to the same level:

```
ciscoasa(config)# privilege level 5 command filter
```

The **show privilege** command separates the forms in the display.

The following example shows the use of the **mode** keyword. The **enable** command must be entered from user EXEC mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level:

```
ciscoasa(config)# privilege cmd level 0 mode enable command enable
ciscoasa(config)# privilege cmd level 15 mode cmd command enable
ciscoasa(config)# privilege show level 15 mode cmd command enable
```

The following example shows an additional command, the **configure** command, which uses the **mode** keyword:

```
ciscoasa(config)# privilege show level 5 mode cmd command configure
ciscoasa(config)# privilege clear level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode enable command configure
```

Note This last line is for the **configure terminal** command.

Configure Commands on the TACACS+ Server

You can configure commands on a Cisco Secure Access Control Server (ACS) TACACS+ server as a shared profile component, for a group, or for individual users. For third-party TACACS+ servers, see your server documentation for more information about command authorization support.

See the following guidelines for configuring commands in Cisco Secure ACS Version 3.1; many of these guidelines also apply to third-party servers:

- The ASA sends the commands to be authorized as shell commands, so configure the commands on the TACACS+ server as shell commands.

Note Cisco Secure ACS might include a command type called "pix-shell." Do not use this type for ASA command authorization.

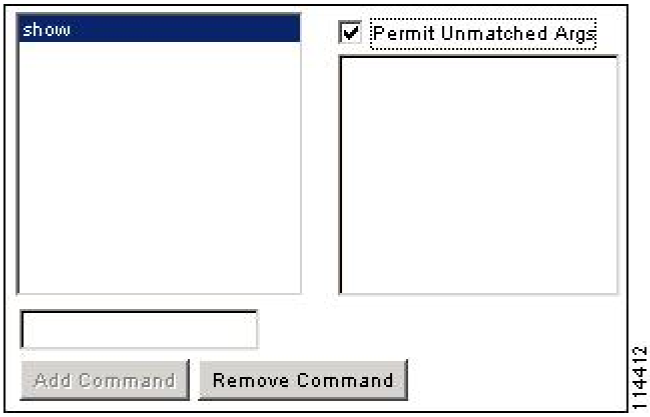
- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

  For example, to allow the **show running-configuration aaa-server** command, add **show running-configuration** to the command field, and type **permit aaa-server** in the arguments field.

- You can permit all arguments of a command that you do not explicitly deny by checking the **Permit Unmatched Args** check box.

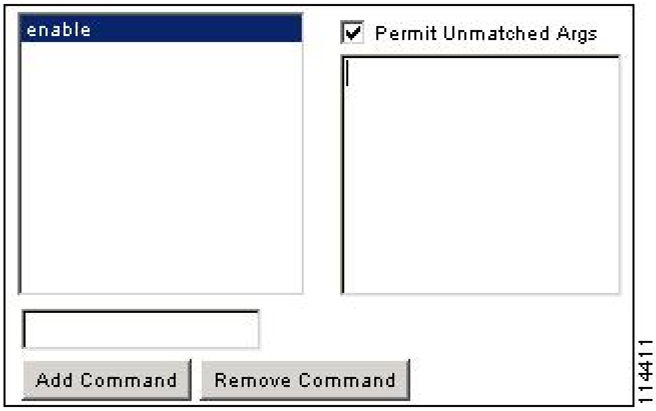
  For example, you can configure just the **show** command, then all the **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and a question mark, which shows CLI usage (see the following figure).

Figure 55: Permitting All Related Commands



- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see the following figure).

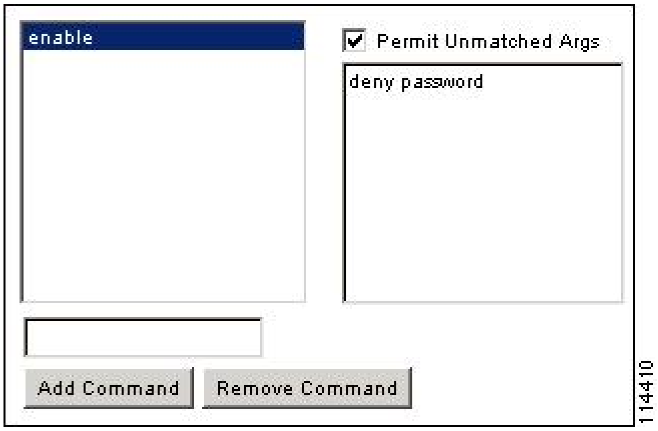
Figure 56: Permitting Single Word Commands



- To disallow some arguments, enter the arguments preceded by **deny**.

For example, to allow **enable**, but not **enable password**, enter **enable** in the commands field, and **deny password** in the arguments field. Be sure to check the **Permit Unmatched Args** check box so that **enable** alone is still allowed (see the following figure).

Figure 57: Disallowing Arguments



enable

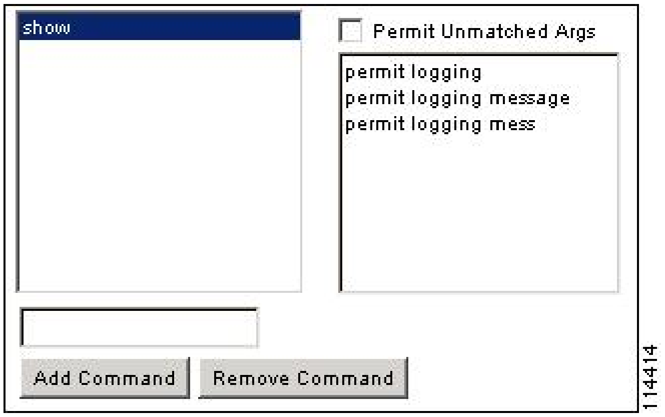
☑ Permit Unmatched Args

deny password

- When you abbreviate a command at the command line, the ASA expands the prefix and main command to the full text, but it sends additional arguments to the TACACS+ server as you enter them.

  For example, if you enter **sh log**, then the ASA sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the ASA sends **show logging mess** to the TACACS+ server, and not the expanded command **show logging message**. You can configure multiple spellings of the same argument to anticipate abbreviations (see the following figure).

Figure 58: Specifying Abbreviations



show

□ Permit Unmatched Args

permit logging
permit logging message
permit logging mess

- We recommend that you allow the following basic commands for all users:
  - **show checksum**
  - **show curpriv**
  - **enable**
  - **help**
  - **show history**

- **login**
- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

Configure TACACS+ Command Authorization

If you enable TACACS+ command authorization, and a user enters a command at the CLI, the ASA sends the command and username to the TACACS+ server to determine if the command is authorized.

Before you enable TACACS+ command authorization, be sure that you are logged into the ASA as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the ASA. For example, you should log in as an admin user with all commands authorized. Otherwise, you could become unintentionally locked out.

Do not save your configuration until you are sure that it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the ASA.

Be sure that your TACACS+ system is completely stable and reliable. The necessary level of reliability typically requires that you have a fully redundant TACACS+ server system and fully redundant connectivity to the ASA. For example, in your TACACS+ server pool, include one server connected to interface 1, and another to interface 2. You can also configure local command authorization as a fallback method if the TACACS+ server is unavailable.

To configure command authorization using a TACACS+ server, perform the following steps:

Procedure

Enter the following command:

aaa authorization command *tacacs+_server_group* [**LOCAL**]

Example:

```
ciscoasa(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the ASA to use the local database as a fallback method if the TACACS+ server is unavailable. To enable fallback, specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the TACACS+ server because the ASA prompt does not give any indication of which method is being used. Be sure to configure users in the local database and command privilege levels.

Configure a Password Policy for Local Database Users

When you configure authentication for CLI or ASDM access using the local database, you can configure a password policy that requires a user to change their password after a specified amount of time and also requires password standards such as a minimum length and the minimum number of changed characters.

The password policy only applies to administrative users using the local database, and not to other types of traffic that can use the local database, such as VPN or AAA for network access, and not to users authenticated by a AAA server.

After you configure the password policy, when you change a password (either your own or another user's), the password policy applies to the new password. Any existing passwords are grandfathered in. The new policy applies to changing the password with the **username** command as well as the **change-password** command.

Before you begin

- Configure AAA authentication for CLI or ASDM access using the local database.
- Specify usernames in the local database.

Procedure

Step 1 (Optional) Set the interval in days after which passwords expire for remote users.

password-policy lifetime *days*

Example:

```
ciscoasa(config)# password-policy lifetime 180
```

Note Users at the console port are never locked out because of password expiration.

Valid values are between 0 and 65536 days. The default value is 0 days, a value indicating that passwords will never expire.

Seven days before the password expires, a warning message appears. After the password expires, system access is denied to remote users. To gain access after expiration, do one of the following:

- Have another administrator change your password with the **username** command.
- Log in to the physical console port to change your password.

Step 2 (Optional) Set the minimum number of characters that you must change between new and old passwords.

password-policy minimum-changes *value*

Example:

```
ciscoasa(config)# password-policy minimum-changes 2
```

Valid values are between 0 and 64 characters. The default value is 0.

Character matching is position independent, meaning that new password characters are considered changed only if they do not appear anywhere in the current password.

Step 3 (Optional) Set the minimum length of passwords.

password-policy minimum-length *value*

Example:

```
ciscoasa(config)# password-policy minimum-length 8
```

Valid values are between 3 and 64 characters. We recommend a minimum password length of 8 characters.

Step 4 (Optional) Set the minimum number of upper case characters that passwords must have.

password-policy minimum-uppercase *value*

Example:

```
ciscoasa(config)# password-policy  minimum-uppercase 3
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 5 (Optional) Set the minimum number of lower case characters that passwords must have.

password-policy minimum-lowercase *value*

Example:

```
ciscoasa(config)# password-policy minimum-lowercase 6
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 6 (Optional) Set the minimum number of numeric characters that passwords must have.

password-policy minimum-numeric *value*

Example:

```
ciscoasa(config)# password-policy minimum-numeric 1
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 7 (Optional) Set the minimum number of special characters that passwords must have.

password-policy minimum-special *value*

Example:

```
ciscoasa(config)# password-policy minimum-special 2
```

Valid values are between 0 and 64 characters. Special characters include the following: !, @, #, $, %, ^, &, *, ‘(‘ and ‘)’. The default value is 0, which means there is no minimum.

Step 8 (Optional) Set whether users must change their password using the **change-password** command, instead of letting users change their password with the **username** command.

password-policy authenticate enable

Example:

```
ciscoasa(config)# password-policy authenticate enable
```

The default setting is disabled: a user can use either method to change their password.

If you enable this feature and try to change your password with the **username** command, the following error message appears:

```
ERROR: Changing your own password is prohibited
```

You also cannot delete your own account with the **clear configure username** command. If you try, the following error message appears:

```
ERROR: You cannot delete all usernames because you are not allowed to delete yourself
```

Change Your Password

If you configure a password lifetime in the password policy, you need to change your password to a new one when the old password expires. This password change method is required if you enable password policy authentication. If password policy authentication is not enabled, then you can use this method, or you can change your user account directly.

To change your username password, perform the following steps:

Procedure

Enter the following command:

change-password [**old-password** *old_password* [**new-password** *new_password*]]

Example:

```
ciscoasa# change-password old-password j0hncr1chton new-password a3rynsun
```

If you do not enter the old and new passwords in the command, the ASA prompts you for input.

Configure Management Access Accounting

You can send accounting messages to the TACACS+ accounting server when you enter any command other than **show** commands at the CLI. You can configure accounting when users log in, when they enter the **enable** command, or when they issue commands.

For command accounting, you can only use TACACS+ servers.

To configure management access and enable command accounting, perform the following steps:

Procedure

Step 1 Enter the following command:

aaa accounting {**serial** | **telnet** | **ssh** | **enable**} **console** *server-tag*

Example:

```
ciscoasa(config)# aaa accounting telnet console group_1
```

Valid server group protocols are RADIUS and TACACS+.

Step 2 Enable command accounting. Only TACACS+ servers support command accounting.

aaa accounting command [**privilege** *level*] *server-tag*

Example:

```
ciscoasa(config)# aaa accounting command privilege 15 group_1
```

The **privilege** *level* keyword-argument pair is the minimum privilege level and the *server-tag* argument is the name of the TACACS+ server group to which the ASA should send command accounting messages.

Recover from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the ASA CLI. You can usually recover access by restarting the ASA. However, if you already saved your configuration, you might be locked out.

The following table lists the common lockout conditions and how you might recover from them.

Table 43: CLI Authentication and Command Authorization Lockout Scenarios

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users have been configured in the local database.	If you have no users in the local database, you cannot log in, and you cannot add any users.	Log in and reset the passwords and **aaa** commands.	Session into the ASA from the switch. From the system execution space, you can change to the context and add a user.

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
TACACS+ command authorization TACACS+ CLI authentication RADIUS CLI authentication	The server is down or unreachable and you do not have the fallback method configured.	If the server is unreachable, then you cannot log in or enter any commands.	1. Log in and reset the passwords and AAA commands. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.	1. If the server is unreachable because the network configuration is incorrect on the ASA, session into the ASA from the switch. From the system execution space, you can change to the context and reconfigure your network settings. 2. Configure the local database as a fallback method so that you do not get locked out when the server is down.
TACACS+ command authorization	You are logged in as a user without enough privileges or as a user that does not exist.	You enable command authorization, but then find that the user cannot enter any more commands.	Fix the TACACS+ server user account. If you do not have access to the TACACS+ server and you need to configure the ASA immediately, then log into the maintenance partition and reset the passwords and **aaa** commands.	Session into the ASA from the switch. From the system execution space, you can change to the context and complete the configuration changes. You can also disable command authorization until you fix the TACACS+ configuration.
Local command authorization	You are logged in as a user without enough privileges.	You enable command authorization, but then find that the user cannot enter any more commands.	Log in and reset the passwords and **aaa** commands.	Session into the ASA from the switch. From the system execution space, you can change to the context and change the user level.

Monitoring Device Access

See the following commands for monitoring device access:

- **show running-config all privilege all**

  This command shows privilege levels for all commands.

  For the **show running-config all privilege all** command, the ASA displays the current assignment of each CLI command to a privilege level. The following is sample output from this command:

```
ciscoasa(config)# show running-config all  privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
...
```

- **show running-config privilege level** *level*

  This command shows commands for a specific privilege level. The level argument is an integer between 0 and 15.

  The following example shows the command assignments for privilege level 10:

  ```
  ciscoasa(config)# show running-config all privilege level 10
  privilege show level 10 command aaa
  ```

- **show running-config privilege command** *command*

  This command shows the privilege level of a specific command.

  The following example shows the command assignments for the **access-list** command:

  ```
  ciscoasa(config)# show running-config all privilege command access-list
  privilege show level 15 command access-list
  privilege clear level 15 command access-list
  privilege configure level 15 command access-list
  ```

- **show curpriv**

  This command shows the currently logged-in user.

  The following is sample output from the **show curpriv** command:

  ```
  ciscoasa# show curpriv
  Username: admin
  Current privilege level: 15
  Current Mode/s: P_PRIV
  ```

  The following table describes the **show curpriv** command output.

  Table 44: show curpriv Command Output Description

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (user EXEC) or enable_15 (privileged EXEC).

Field	Description
Current privilege level	Levels range from 0 to 15. Unless you configure local command authorization and assign commands to intermediate privilege levels, levels 0 and 15 are the only levels that are used.
Current Modes	The available access modes are the following: • P_UNPR—User EXEC mode (levels 0 and 1) • P_PRIV—Privileged EXEC mode (levels 2 to 15) • P_CONF—Configuration mode

- **show quota management-session**

  This command shows the current sessions in use.

  The following is sample output from the **show quota management-session** command:

```
ciscoasa(config)# show quota management-session

quota management-session limit 3
quota management-session warning level 2
quota management-session level 0
quota management-session high water 2
quota management-session errors 0
quota management-session warnings 0
```

History for Management Access

Table 45: History for Management Access

Feature Name	Platform Releases	Description
ASDM management authorization	9.4(1)	You can now configure management authorization separately for HTTP access vs. Telnet and SSH access. We introduced the following command: **aaa authorization http console**
ASDM username from certificate configuration	9.4(1)	When you enable ASDM certificate authentication (**http authentication-certificate**), you can configure how ASDM extracts the username from the certificate; you can also enable pre-filling the username at the login prompt. We introduced the following command: **http username-from-certificate**

Feature Name	Platform Releases	Description
Improved one-time password authentication	9.2(1)	Administrators who have sufficient authorization privileges may enter privileged EXEC mode by entering their authentication credentials once. The **auto-enable** option was added to the **aaa authorization exec** command. We modified the following command: **aaa authorization exec**.
Configurable SSH encryption and integrity ciphers	9.1(7)/9.4(3)	Users can select cipher modes when doing SSH encryption management and can configure HMAC and encryption for varying key exchange algorithms. You might want to change the ciphers to be more or less strict, depending on your application. Note that the performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use **ssh cipher encryption custom aes128-cbc**, for example. We introduced the following commands: **ssh cipher encryption, ssh cipher integrity**.
AES-CTR encryption for SSH	9.1(2)	The SSH server implementation in the ASA now supports AES-CTR mode encryption.
Improved SSH rekey interval	9.1(2)	An SSH connection is rekeyed after 60 minutes of connection time or 1 GB of data traffic. We introduced the following command: **show ssh sessions detail**.
For the ASASM in multiple context mode, support for Telnet and virtual console authentication from the switch.	8.5(1)	Although connecting to the ASASM from the switch in multiple context mode connects to the system execution space, you can configure authentication in the admin context to govern those connections.
Support for administrator password policy when using the local database	8.4(4.1), 9.1(2)	When you configure authentication for CLI or ASDM access using the local database, you can configure a password policy that requires a user to change their password after a specified amount of time and also requires password standards such as a minimum length and the minimum number of changed characters. We introduced the following commands: **change-password**, **password-policy lifetime**, **password-policy minimum changes**, **password-policy minimum-length**, **password-policy minimum-lowercase**, **password-policy minimum-uppercase**, **password-policy minimum-numeric**, **password-policy minimum-special**, **password-policy authenticate enable**, **clear configure password-policy**, **show running-config password-policy**.

Feature Name	Platform Releases	Description
Support for SSH public key authentication	8.4(4.1), 9.1(2)	You can enable public key authentication for SSH connections to the ASA on a per-user basis. You can specify a public key file (PKF) formatted key or a Base64 key. The PKF key can be up to 4096 bits. Use PKF format for keys that are too large to for the ASA support of the Base64 format (up to 2048 bits). We introduced the following commands: **ssh authentication**. *PKF key format support is only in 9.1(2) and later.*
Support for Diffie-Hellman Group 14 for the SSH Key Exchange	8.4(4.1), 9.1(2)	Support for Diffie-Hellman Group 14 for SSH Key Exchange was added. Formerly, only Group 1 was supported. We introduced the following command: **ssh key-exchange**.
Support for a maximum number of management sessions	8.4(4.1), 9.1(2)	You can set the maximum number of simultaneous ASDM, SSH, and Telnet sessions. We introduced the following commands: **quota management-session**, **show running-config quota management-session**, **show quota management-session**.
Increased SSH security; the SSH default username is no longer supported.	8.4(2)	Starting in 8.4(2), you can no longer connect to the ASA using SSH with the pix or asa username and the login password. To use SSH, you must configure AAA authentication using the **aaa authentication ssh console LOCAL** command (CLI) or Configuration > Device Management > Users/AAA > AAA Access > Authentication (ASDM); then define a local user by entering the **username** command (CLI) or choosing Configuration > Device Management > Users/AAA > User Accounts (ASDM). If you want to use a AAA server for authentication instead of the local database, we recommend also configuring local authentication as a backup method.
Management Access	7.0(1)	We introduced this feature. We introduced the following commands: **show running-config all privilege all**, **show running-config privilege level**, **show running-config privilege command**, **telnet**, **telnet timeout**, **ssh**, **ssh timeout**, **http**, **http server enable**, **asdm image disk, banner**, **console timeout**, **icmp**, **ipv6 icmp**, **management access**, **aaa authentication console**, **aaa authentication enable console**, **aaa authentication telnet | ssh console**, **service-type**, **login**, **privilege**, **aaa authentication exec authentication-server**, **aaa authentication command LOCAL**, **aaa accounting serial | telnet | ssh | enable console**, **show curpriv**, **aaa accounting command privilege**.

CHAPTER 37

Software and Configurations

This chapter describes how to manage the Cisco ASA software and configurations.

Upgrade the Software

See the Cisco ASA Upgrade Guide for full upgrade procedures.

Load an Image Using ROMMON

To load a software image onto an ASA from the ROMMON mode using TFTP, perform the following steps.

Procedure

Step 1 Connect to the ASA console port according to the instructions in Access the Appliance Console, on page 19.

Step 2 Power off the ASA, then power it on.

Step 3 During startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 In ROMMOM mode, define the interface settings to the ASA, including the IP address, TFTP server address, gateway address, software image file, and port, as follows:

```
rommon #1> interface gigabitethernet0/0
rommon #2> address 10.86.118.4
rommon #3> server 10.86.118.21
rommon #4> gateway 10.86.118.21
rommon #5> file asa961-smp-k8.bin
```

Note Be sure that the connection to the network already exists.

The **interface** command is ignored on the ASA 5506-X, ASA 5508-X, and ASA 5516-X platforms, and you must perform TFTP recovery on these platforms from the Management 1/1 interface.

Step 5 Validate your settings:

```
rommon #6> set
ROMMON Variable Settings:
  ADDRESS=10.86.118.3
  SERVER=10.86.118.21
  GATEWAY=10.86.118.21
  PORT=GigabitEthernet0/0
  VLAN=untagged
  IMAGE=asa961-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20
```

Step 6 Ping the TFTP server:

```
rommon #7> ping server
Sending 20, 100-byte ICMP Echoes to server 10.86.118.21, timeout is 4 seconds:

Success rate is 100 percent (20/20)
```

Step 7 Save the network settings for future use:

```
rommon #8> sync
Updating NVRAM Parameters...
```

Step 8 Load the software image:

```
rommon #9> tftpdnld
ROMMON Variable Settings:
  ADDRESS=10.86.118.3
  SERVER=10.86.118.21
  GATEWAY=10.86.118.21
  PORT=GigabitEthernet0/0
  VLAN=untagged
  IMAGE=asa961-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20

tftp asa961-smp-k8.bin@10.86.118.21 via 10.86.118.21

Received 14450688 bytes

Launching TFTP Image...
Cisco ASA Security Appliance admin loader (3.0) #0: Mon Mar 5 16:00:07 MST 2016
```

```
Loading...
```

After the software image is successfully loaded, the ASA automatically exits ROMMON mode.

Step 9 Booting the ASA from ROMMON mode does not preserve the system image across reloads; you must still download the image to flash memory. See Upgrade the Software, on page 875.

Load an Image for the ASASM Using ROMMON

To load a software image to an ASASM from the ROMMON mode using TFTP, perform the following steps.

Procedure

Step 1 Connect to the ASA console port according to the instructions in Access the ASA Services Module Console, on page 21.

Step 2 Make sure that you reload the ASASM image.

Step 3 During startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 In ROMMOM mode, define the interface settings to the ASASM, including the IP address, TFTP server address, gateway address, software image file, port, and VLAN, as follows:

```
rommon #2> address 10.86.118.4
rommon #3> server 10.86.118.21
rommon #4> gateway 10.86.118.21
rommon #5> file asa961-smp-k8.bin
rommon #5> interface Data0
rommon #6> vlan 1
Data0
Link is UP
MAC Address: 0012.d949.15b8
```

Note Be sure that the connection to the network already exists.

Step 5 Validate your settings:

```
rommon #7> set
ROMMON Variable Settings:
  ADDRESS=10.86.118.4
  SERVER=10.86.118.21
  GATEWAY=10.86.118.21
  PORT=Data0
  VLAN=1
  IMAGE=asa961-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=2
  RETRY=20
```

Step 6 Ping the TFTP server:

```
rommon #8> ping server
Sending 20, 100-byte ICMP Echoes to server 10.86.118.21, timeout is 2 seconds:

Success rate is 100 percent (20/20)
```

Step 7 Load the software image:

```
rommon #9> tftpdnld
Clearing EOBC receive queue ...
cmostime_set = 1
ROMMON Variable Settings:
  ADDRESS=10.86.118.3
  SERVER=10.86.118.21
  GATEWAY=10.86.118.21
  PORT=Data0
  VLAN=1
  IMAGE=asa961-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20

tftp asa961-smp-k8.bin@10.86.118.21 via 10.86.118.21
Starting download. Press ESC to abort.
```

After the software image is successfully loaded, the ASASM automatically exits ROMMON mode.

Step 8 Booting the module from ROMMON mode does not preserve the system image across reloads; you must still download the image to flash memory. See Upgrade the Software, on page 875.

Upgrade the ROMMON Image (ASA 5506-X, 5508-X, and 5516-X, ISA 3000)

Follow these steps to upgrade the ROMMON image for the ASA 5506-X series, ASA 5508-X, ASA 5516-X, and ISA 3000. For the ASA models, the ROMMON version on your system must be 1.1.8 or greater. We recommend that you upgrade to the latest version.

You can only upgrade to a new version; you cannot downgrade.

Caution The ASA 5506-X, 5508-X, and 5516-X ROMMON upgrade for 1.1.15 and the ISA 3000 ROMMON upgrade for 1.0.5 takes twice as long as previous ROMMON versions, approximately 15 minutes. **Do not** power cycle the device during the upgrade. If the upgrade is not complete within 30 minutes or it fails, contact Cisco technical support; **do not** power cycle or reset the device.

Before you begin

Obtain the new ROMMON image from Cisco.com, and put it on a server to copy to the ASA. The ASA supports FTP, TFTP, SCP, HTTP(S), and SMB servers. Download the image from:

- ASA 5506-X, 5508-X, 5516-X: https://software.cisco.com/download/home/286283326/type
- ISA 3000: https://software.cisco.com/download/home/286288493/type

Procedure

Step 1 Copy the ROMMON image to the ASA flash memory. This procedure shows an FTP copy; enter **copy ?** for the syntax for other server types.

copy ftp://[*username*:*password*@]*server_ip*/**asa5500-firmware-***xxxx***.SPA disk0:asa5500-firmware-***xxxx***.SPA**

Step 2 To see your current version, enter the **show module** command and look at the Fw Version in the output for Mod 1 in the MAC Address Range table:

```
ciscoasa# show module
[...]
Mod  MAC Address Range                 Hw Version   Fw Version   Sw Version
---- --------------------------------- ------------ ------------ ---------------
   1 7426.aceb.ccea to 7426.aceb.ccf2  0.3          1.1.5        9.4(1)
 sfr 7426.aceb.cce9 to 7426.aceb.cce9  N/A          N/A
```

Step 3 Upgrade the ROMMON image:

upgrade rommon disk0:asa5500-firmware-*xxxx***.SPA**

Example:

```
ciscoasa# upgrade rommon disk0:asa5500-firmware-1108.SPA
Verifying file integrity of disk0:/asa5500-firmware-1108.SPA

Computed Hash   SHA2: d824bdeecee1308fc64427367fa559e9
                      eefe8f182491652ee4c05e6e751f7a4f
                      5cdea28540cf60acde3ab9b65ff55a9f
                      4e0cfb84b9e2317a856580576612f4af

Embedded Hash   SHA2: d824bdeecee1308fc64427367fa559e9
                      eefe8f182491652ee4c05e6e751f7a4f
                      5cdea28540cf60acde3ab9b65ff55a9f
                      4e0cfb84b9e2317a856580576612f4af


Digital signature successfully validated
File Name                     : disk0:/asa5500-firmware-1108.SPA
Image type                    : Release
    Signer Information
        Common Name           : abraxas
        Organization Unit     : NCS_Kenton_ASA
        Organization Name     : CiscoSystems
    Certificate Serial Number : 553156F4
    Hash Algorithm            : SHA2 512
    Signature Algorithm       : 2048-bit RSA
    Key Version               : A
Verification successful.
Proceed with reload? [confirm]
```

Step 4 Confirm to reload the ASA when you are prompted.

The ASA upgrades the ROMMON image, and then reloads the operating system.

Recover and Load an Image for the ASA 5506W-X Wireless Access Point

To recover and load a software image onto an ASA 5506W-X using TFTP, perform the following steps.

Procedure

Step 1 Session to the access point (AP) and enter the AP ROMMON (not the ASA ROMMON):

```
ciscoasa# hw-module module wlan recover image
```

Step 2 Follow the procedure in the Cisco IOS Software Configuration Guide for Cisco Aironet Access Points.

Downgrade Your Software

In many cases, you can downgrade your ASA software and restore a backup configuration from the previous software version. The method of downgrading depends on your ASA platform.

Guidelines and Limitations for Downgrading

See the following guidelines before downgrading:

- There is no official Zero Downtime Downgrade support for clustering. However, in some cases, Zero Downtime Downgrading will work. See the following known issues for downgrading; note that there may be other issues that require you to reload your cluster units, thus causing downtime.
  - Downgrade from 9.2(1) or later to 9.1 or earlier with clustering—Zero Downtime Downgrade is not supported.
- Downgrade to 9.5 and earlier with passwords using PBKDF2 (Password-Based Key Derivation Function 2) hash—Versions before 9.6 do not support PBKDF2 hashing. In 9.6(1), **enable** and **username** passwords longer than 32 characters use PBKDF2 hashing. If you downgrade, the **enable** password reverts to the default (which is blank). Usernames will not parse correctly, and the **username** commands will be removed. You must re-create your local users.
- Downgrade from Version 9.5(2.200) for the ASAv—The ASAv does not retain the licensing registration state. You need to re-register with the **license smart register idtoken id_token force** command (for ASDM: see the **Configuration > Device Management > Licensing > Smart Licensing** page, and use the **Force registration** option); obtain the ID token from the Smart Software Manager.

- VPN tunnels are replicated to the standby unit even if the standby unit is running a version of software that does not support the Ciphersuite that the original tunnel negotiated. This scenario occurs when downgrading. In this case, disconnect your VPN connection and reconnect.

Incompatible Configuration Removed After Downgrading

When you downgrade to an old version, commands that were introduced in later versions will be removed from the configuration. There is no automated way to check the configuration against the target version before you downgrade. You can view when new commands were added in ASA new features by release.

You can view rejected commands *after* you downgrade using the **show startup-config errors** command. If you can perform a downgrade on a lab device, you can preview the effects using this command before you perform the downgrade on a production device.

In some cases, the ASA migrates commands to new forms automatically when you upgrade, so depending on your version, even if you did not manually configure new commands, the downgrade could be affected by configuration migrations. We recommend that you have a backup of your old configuration that you can use when you downgrade. In the case of upgrading to 8.3, a backup is automatically created (<old_version>_startup_cfg.sav). Other migrations do not create back-ups. See the "Version-Specific Guidelines and Migrations" in the ASA Upgrade guide for more information about automatic command migrations that could affect downgrading.

See also known downgrade issues in Guidelines and Limitations for Downgrading, on page 880.

For example, an ASA running version 9.8(2) includes the following commands:

```
access-list acl1 extended permit sctp 192.0.2.0 255.255.255.0 198.51.100.0 255.255.255.0
username test1 password $sha512$1234$abcdefghijklmnopqrstuvwxyz privilege 15
snmp-server user snmpuser1 snmpgroup1 v3 engineID abcdefghijklmnopqrstuvwxyz encrypted auth
 md5 12:ab:34 priv aes 128 12:ab:34
```

When you downgrade to 9.0(4), you will see the following errors on startup:

```
access-list acl1 extended permit sctp 192.0.2.0 255.255.255.0 198.51.100.0 255.255.255.0
                                 ^
ERROR: % Invalid input detected at '^' marker.

username test1 password $sha512$1234$abcdefghijklmnopqrstuvwxyz pbkdf2 privilege 15
                                                                ^
ERROR: % Invalid input detected at '^' marker.

snmp-server user snmpuser1 snmpgroup1 v3 engineID abcdefghijklmnopqrstuvwxyz encrypted auth
 md5 12:ab:34 priv aes 128 12:ab:34
                                         ^
ERROR: % Invalid input detected at '^' marker.
```

In this example, support for **sctp** in the **access-list extended** command was added in version 9.5(2), support for **pbkdf2** in the **username** command was added in version 9.6(1), and support for **engineID** in the **snmp-server user** command was added in version 9.5(3).

Downgrade the Firepower 4100/9300

You can downgrade the ASA software version by restoring the backup configuration to the startup configuration, setting the ASA version to the old version, and then reloading.

Before you begin

- This procedure requires a backup configuration of the ASA before you upgraded, so you can restore the old configuration. If you do not restore the old configuration, you may have incompatible commands representing new or changed features. Any new commands will be rejected when you load the old software version.
- Make sure the old ASA version is compatibile with the current FXOS version. If not, downgrade FXOS as the first step before you restore the old ASA configuration. Just make sure the downgraded FXOS is also compatible with the current ASA version (before you downgrade it). If you cannot achieve compatibility, we suggest you do not perform a downgrade.

Procedure

Step 1 At the ASA CLI, copy the backup ASA configuration to the startup configuration. For failover or clustering, perform this step on the active/control unit. This step replicates the command to the standby/data units.

copy *old_config_url* **startup-config**

It's important that you do not save the running configuration to the startup configuration using **write memory**; this command will overwrite your backup configuration.

Example:

```
ciscoasa# copy disk0:/9.8.4_cfg.sav startup-config
```

Step 2 In FXOS, use the Firepower Chassis Manager or FXOS CLI to use the old ASA software version using the upgrade procedure in the ASA upgrade guide for standalone, failover, or clustering deployments. In this case, specify the old ASA version instead of a new version.

Step 3 If you are also downgrading FXOS, use the Firepower Chassis Manager or FXOS CLI to set the old FXOS software version to be the current version using the upgrade procedure in the ASA upgrade guide for standalone, failover, or clustering deployments.

Downgrade the ASA 5500-X or ISA 3000

The downgrade feature provides a shortcut for completing the following functions on ASA 5500-X and ISA 3000 models:

- Clearing the boot image configuration (**clear configure boot**).
- Setting the boot image to be the old image (**boot system**).
- (Optional) Entering a new activation key (**activation-key**).
- Saving the running configuration to startup (**write memory**). This sets the BOOT environment variable to the old image, so when you reload, the old image is loaded.
- Copying the old configuration backup to the startup configuration (**copy** *old_config_url* **startup-config**).
- Reloading (**reload**).

Before you begin

- This procedure requires a backup configuration of the ASA before you upgraded, so you can restore the old configuration.
- Make sure the ASA FirePOWER module version, if installed, is compatible with the old ASA version. You cannot downgrade the FirePOWER module to an earlier major version.

Procedure

ASA 5500-X models only: Downgrade the software and restore the old configuration.

downgrade [**/noconfirm**] *old_image_url old_config_url* [**activation-key** *old_key*]

Example:

```
ciscoasa(config)# downgrade /noconfirm disk0:/asa821-k8.bin disk0:/8_2_1_0_startup_cfg.sav
```

The **/noconfirm** option downgrades without prompting. The *image_url* is the path to the old image on disk0, disk1, tftp, ftp, or smb. The *old_config_url* is the path to the saved, pre-migration configuration. If you need to revert to a pre-8.3 activation key, then you can enter the old activation key.

Manage Files

View Files in Flash Memory

You can view files in flash memory and see information about files.

Procedure

Step 1 View files in flash memory:

dir [**disk0:** | **disk1:**]

Example:

```
hostname# dir

Directory of disk0:/
500    -rw-  4958208     22:56:20 Nov 29 2004  cdisk.bin
2513   -rw-  4634        19:32:48 Sep 17 2004  first-backup
2788   -rw-  21601       20:51:46 Nov 23 2004  backup.cfg
2927   -rw-  8670632     20:42:48 Dec 08 2004  asdmfile.bin
```

Enter **disk0:** for the internal flash memory. The **disk1:** keyword represents the external flash memory. The internal flash memory is the default.

Step 2 View extended information about a specific file:

show file information [path:/]*filename*

Example:

```
hostname# show file information cdisk.bin

disk0:/cdisk.bin:
  type is image (XXX) []
  file size is 4976640 bytes version 7.0(1)
```

The file size listed is for example only.

The default path is the root directory of the internal flash memory (disk0:/).

Delete Files from Flash Memory

You can remove files from flash memory that you no longer need.

Procedure

Delete a file from flash memory:

delete disk0: *filename*

By default, the file is deleted from the current working directory if you do not specify a path. You may use wildcards when deleting files. You are prompted with the filename to delete, and then you must confirm the deletion.

Erase the Flash File System

To erase the flash file system, perform the following steps.

Procedure

Step 1 Connect to the ASA console port according to the instructions in Access the ASA Services Module Console, on page 21 or Access the Appliance Console, on page 19.

Step 2 Power off the ASA, then power it on.

Step 3 During startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 Enter the **erase** command, which overwrites all files and erases the file system, including hidden system files:

rommon #1> **erase** [**disk0:** | **disk1:** | **flash:**]

Configure File Access

The ASA can use an FTP client, secure copy client, or TFTP client. You can also configure the ASA as a secure copy server so you can use a secure copy client on your computer.

Configure the FTP Client Mode

The ASA can use FTP to upload or download image files or configuration files to or from an FTP server. In passive FTP, the client initiates both the control connection and the data connection. The server, which is the recipient of the data connection in passive mode, responds with the port number to which it is listening for the specific connection.

Procedure

Set the FTP mode to passive:

ftp mode passive

Example:

```
ciscoasa(config)# ftp mode passive
```

Configure the ASA as a Secure Copy Server

You can enable the secure copy (SCP) server on the ASA. Only clients that are allowed to access the ASA using SSH can establish a secure copy connection.

Before you begin

- The server does not have directory support. The lack of directory support limits remote client access to the ASA internal files.
- The server does not support banners or wildcards.
- Enable SSH on the ASA according to Configure SSH Access, on page 837.
- The ASA license must have the strong encryption (3DES/AES) license to support SSH Version 2 connections.
- Unless otherwise specified, for multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.
- The performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use the **ssh cipher encryption** command; for example, **ssh cipher encryption custom aes128-cbc**

Procedure

Step 1 Enable the SCP server:

ssh scopy enable

Step 2 (Optional) Manually add or delete servers and their keys from the ASA database:

ssh pubkey-chain [**no**] **server** *ip_address* {**key-string** *key_string* **exit**| **key-hash** {**md5** | **sha256**} *fingerprint*}

Example:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit
ciscoasa(config-ssh-pubkey-server)# show running-config ssh pubkey-chain
ssh pubkey-chain
  server 10.7.8.9
    key-hash sha256 f1:22:49:47:b6:76:74:b2:db:26:fb:13:65:d8:99:19:
e7:9e:24:46:59:be:13:7f:25:27:70:9b:0e:d2:86:12
```

The ASA stores the SSH host key for each SCP server to which it connects. You can manually manage keys if desired.

For each server, you can specify the **key-string** (public key) or **key-hash** (hashed value) of the SSH host.

The *key_string* is the Base64 encoded RSA public key of the remote peer. You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file. After you submit the Base64 encoded public key, that key is then hashed via SHA-256.

The **key-hash** {**md5** | **sha256**} *fingerprint* enters the already hashed key (using an MD5 or SHA-256 key); for example, a key that you copied from **show** command output.

Step 3 (Optional) Enable or disable SSH host key checking. For multiple context mode, enter this command in the admin context.

[**no**] **ssh stricthostkeycheck**

Example:

```
ciscoasa# ssh stricthostkeycheck
ciscoasa# copy x scp://cisco@10.86.95.9/x
The authenticity of host '10.86.95.9 (10.86.95.9)' can't be established.
RSA key fingerprint is dc:2e:b3:e4:e1:b7:21:eb:24:e9:37:81:cf:bb:c3:2a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.86.95.9' (RSA) to the list of known hosts.
Source filename [x]?

Address or name of remote host [10.86.95.9]?

Destination username [cisco]?

Destination password []? cisco123

Destination filename [x]?
```

By default, this option is enabled. When this option is enabled, you are prompted to accept or reject the host key if it is not already stored on the ASA. When this option is disabled, the ASA accepts the host key automatically if it was not stored before.

Examples

From a client on the external host, perform an SCP file transfer. For example, in Linux enter the following command:

scp -v -pw *password* [*path*/]*source_filename*
username@*asa_address***:**{**disk0**|**disk1**}**:/**[*path*/]*dest_filename*

The **-v** is for verbose, and if **-pw** is not specified, you will be prompted for a password.

The following example adds an already hashed host key for the server at 10.86.94.170:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.86.94.170
ciscoasa(config-ssh-pubkey-server)# key-hash sha256 65:d9:9d:fe:1a:bc:61:aa:
64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19
```

The following example adds a host string key for the server at 10.7.8.9:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:
46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit
```

Configure the ASA TFTP Client Path

TFTP is a simple client/server file transfer protocol, which is described in RFC 783 and RFC 1350 Rev. 2. You can configure the ASA as a TFTP client so that it can copy files to or from a TFTP server. In this way, you can back up and propagate configuration files to multiple ASAs.

This section lets you predefine the path to a TFTP server so you do not need to enter it in commands such as **copy** and **configure net**.

Procedure

Predefine the TFTP server address and filename for use with **configure net** and **copy** commands:

tftp-server *interface_name server_ip filename*

Example:

```
ciscoasa(config)# tftp-server inside 10.1.4.7 files/config1.cfg
ciscoasa(config)# copy tftp: test.cfg
```

```
Address or name of remote host [10.1.4.7]?

Source filename [files/config1.cfg]?config2.cfg

Destination filename [test.cfg]?

Accessing tftp://10.1.4.7/files/config2.cfg;int=outside...
```

You can override the filename when you enter the command; for example, when you use the **copy** command, you can take advantage of the predefined TFTP server address but still enter any filename at the interactive prompts.

For the **copy** command, enter **tftp:** to use the tftp-server value instead of **tftp://***url*.

Copy a File to the ASA

This section describes how to copy the application image, ASDM software, a configuration file, or any other file that needs to be downloaded to internal or external flash memory from a TFTP, FTP, SMB, HTTP, HTTPS, or SCP server.

Before you begin

- For the IPS SSP software module, before you download the IPS software to disk0, make sure at least 50% of the flash memory is free. When you install IPS, IPS reserves 50% of the internal flash memory for its file system.
- You cannot have two files with the same name but with different letter case in the same directory in flash memory. For example, if you attempt to download the file, Config.cfg, to a location that contains the file, config.cfg, you receive the following error message:

  ```
  %Error opening disk0:/Config.cfg (File exists)
  ```

- For information about installing the Cisco SSL VPN client, see the *Cisco AnyConnect VPN Client Administrator Guide*. For information about installing Cisco Secure Desktop on the ASA, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administrators*.
- To configure the ASA to use a specific application image or ASDM image if you have more than one installed, or have installed them in external flash memory, see Set the ASA Image, ASDM, and Startup Configuration, on page 892.
- For multiple context mode, you must be in the system execution space.

Procedure

Copy a file using one of the following server types.

- Copy from a TFTP server:

  copy [**/noconfirm**] **tftp://***server*[/*path*]/*src_filename* {**disk0**|**disk1**}**:/**[*path*/]*dest_filename*

Example:

```
ciscoasa# copy tftp://10.1.1.67/files/context1.cfg disk0:/context1.cfg

Address or name of remote host [10.1.1.67]?

Source filename [files/context1.cfg]?

Destination filename [context1.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an FTP server:

  copy [**/noconfirm**] **ftp://**[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename* {**disk0**|**disk1**}**:/**[*path*/]*dest_filename*

  Example:

```
ciscoasa# copy ftp://jcrichton:aeryn@10.1.1.67/files/context1.cfg
disk0:/contexts/context1.cfg

Address or name of remote host [10.1.1.67]?

Source username [jcrichton]?

Source password [aeryn]?

Source filename [files/context1.cfg]?

Destination filename [contexts/context1.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an HTTP(S) server:

  copy [**/noconfirm**] **http**[**s**]**://**[*user*[**:***password*]**@**]*server*[**:***port*][/*path*]/*src_filename* {**disk0**|**disk1**}**:/**[*path*/]*dest_filename*

  Example:

```
ciscoasa# copy https://asun:john@10.1.1.67/files/moya.cfg disk0:/contexts/moya.cfg

Address or name of remote host [10.1.1.67]?

Source username [asun]?

Source password [john]?

Source filename [files/moya.cfg]?

Destination filename [contexts/moya.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an SMB server:

copy [**/noconfirm**] **smb:**//[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename*
{**disk0**|**disk1**}**:/**[*path/*]*dest_filename*

Example:

```
ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/test.xml disk0:/test.xml

Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from a SCP server:

  The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the SCP server.

  copy [**/noconfirm**] **scp:**//[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename*[**;int=***interface_name*]
  {**disk0**|**disk1**}**:/**[*path/*]*dest_filename*

  Example:

```
ciscoasa# copy scp://pilot@10.86.94.170/test.cfg disk0:/test.cfg

Address or name of remote host [10.86.94.170]?

Source username [pilot]?

Destination filename [test.cfg]?

The authenticity of host '10.86.94.170 (10.86.94.170)' can't be established.
RSA key fingerprint is
<65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19>(SHA256).
Are you sure you want to continue connecting (yes/no)? yes

Please use the following commands to add the hash key to the configuration:
  ssh pubkey-chain
    server 10.86.94.170
      key-hash sha256
65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19


Password: <type in password>
!!!!!!
6006 bytes copied in 8.160 secs (750 bytes/sec)
```

Copy a File to the Startup or Running Configuration

You can download a text file to the running or startup configuration from a TFTP, FTP, SMB, HTTP(S), or SCP server, or from the flash memory.

Before you begin

When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

Procedure

To copy a file to the startup configuration or running configuration, enter one of the following commands for the appropriate download server:

- Copy from a TFTP server:

  copy [**/noconfirm**] **tftp:**//*server*[/*path*]/*src_filename* {**startup-config** | **running-config**}

  Example:

  ```
  ciscoasa# copy tftp://10.1.1.67/files/old-running.cfg running-config
  ```

- Copy from an FTP server:

  copy [**/noconfirm**] **ftp:**//[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename* {**startup-config** | **running-config**}

  Example:

  ```
  ciscoasa# copy ftp://jcrichton:aeryn@10.1.1.67/files/old-startup.cfg startup-config
  ```

- Copy from an HTTP(S) server:

  copy [**/noconfirm**] **http**[**s**]**:**//[*user*[**:***password*]**@**]*server*[**:***port*][/*path*]/*src_filename* {**startup-config** | **running-config**}

  Example:

  ```
  ciscoasa# copy https://asun:john@10.1.1.67/files/new-running.cfg running-config
  ```

- Copy from an SMB server:

  copy [**/noconfirm**] **smb:**//[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename* {**startup-config** | **running-config**}

  Example:

  ```
  ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/new-running.cfg running-config
  ```

- Copy from a SCP server:

  copy [**/noconfirm**] **scp:**//[*user*[**:***password*]**@**]*server*[/*path*]/*src_filename*[**;int=***interface_name*] {**startup-config** | **running-config**}

  Example:

  ```
  ciscoasa# copy scp://pilot:moya@10.86.94.170/new-startup.cfg startup-config
  ```

The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the SCP server.

Examples

For example, to copy the configuration from a TFTP server, enter the following command:

```
ciscoasa# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
ciscoasa# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
ciscoasa# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Set the ASA Image, ASDM, and Startup Configuration

If you have more than one ASA or ASDM image, you should specify the image that you want to boot. If you do not set the image, the default boot image is used, and that image may not be the one intended. For the startup configuration, you can optionally specify a configuration file.

See the following model guidelines:

- Firepower 9300 chassis—ASA upgrades are managed by FXOS; you cannot upgrade the ASA within the ASA operating system, so do not use this procedure for the ASA image. You can upgrade ASA and FXOS separately from each other, and they are listed separately in the FXOS directory listing. The ASA package always includes ASDM.
- Firepower 2100—The ASA, ASDM, and FXOS images are bundled together into a single package. Package updates are managed by FXOS; you cannot upgrade the ASA within the ASA operating system, so do not use this procedure for the ASA image. You *cannot* upgrade ASA and FXOS separately from each other; they are always bundled together.
- ASDM for the Firepower models—ASDM can be upgraded from within the ASA operating system, so you do not need to only use the bundled ASDM image. ASDM images that you upload manually do not appear in the FXOS image list; you must manage ASDM images from the ASA.

Note When you upgrade the ASA bundle, the ASDM image in the bundle replaces the previous ASDM bundle image on the ASA because they have the same name (**asdm.bin**). But if you manually chose a different ASDM image that you uploaded (for example, **asdm-782.bin**), then you continue to use that image even after a bundle upgrade. To make sure that you are running a compatible version of ASDM, you should either upgrade ASDM before you upgrade the bundle, or you should reconfigure the ASA to use the bundled ASDM image (**asdm.bin**) just before upgrading the ASA bundle.

- ASAv—The initial deployment ASAv package puts the ASA image in the read-only boot:/ partition. When you upgrade the ASAv, you specify a different image in flash memory. Note that if you later clear your configuration (**clear configure all**), then the ASAv will revert to loading the original deployment image. The initial deployment ASAv package also includes an ASDM image that it places in flash memory. You can upgrade the ASDM image separately.

See the following default settings:

- ASA image:
    - Physical ASAs—Boots the first application image that it finds in internal flash memory.
    - ASAv—Boots the image in the read-only boot:/ partition that was created when you first deployed.
    - Firepower 9300 chassis—The FXOS system determines which ASA image to boot. You cannot use this procedure to set the ASA image.
- ASDM image on all ASAs—Boots the first ASDM image that it finds in internal flash memory, or if one does not exist in this location, then in external flash memory.
- Startup configuration—By default, the ASA boots from a startup configuration that is a hidden file.

Procedure

Step 1 Set the ASA boot image location:

boot system *url*

Example:

```
ciscoasa(config)# boot system disk0:/images/asa921.bin
```

The URL can be:

- {**disk0:/** | **disk1:/**}[*path/*]*filename*
- **tftp://**[*user*[**:***password*]**@**]*server*[**:***port*]/[*path/*]*filename*

  The TFTP option is not supported on all models.

You can enter up to four **boot system** command entries to specify different images to boot from in order; the ASA boots the first image it finds successfully. When you enter the **boot system** command, it adds an entry at the bottom of the list. To reorder the boot entries, you must remove all entries using the the **clear configure boot system** command, and re-enter them in the order you desire. Only one **boot system tftp** command can be configured, and it must be the first one configured.

Note If the ASA is stuck in a cycle of constant booting, you can reboot the ASA into ROMMON mode. For more information about the ROMMON mode, see View Debugging Messages, on page 934.

Step 2 Set the ASDM image to boot:

asdm image {**disk0:/** | **disk1:/**}[*path/*]*filename*

Example:

```
ciscoasa(config)# asdm image disk0:/images/asdm721.bin
```

If you do not specify the image to boot, even if you have only one image installed, then the ASA inserts the **asdm image** command into the running configuration. To avoid problems with Auto Update (if configured), and to avoid the image search at each startup, you should specify the ASDM image that you want to boot in the startup configuration.

Step 3 (Optional) Set the startup configuration to be a known file instead of the default hidden file:

boot config {**disk0:/** | **disk1:/**}[*path/*]*filename*

Example:

```
ciscoasa(config)# boot config disk0:/configs/startup1.cfg
```

Back Up and Restore Configurations or Other Files

We recommend that you make regular backups of your configuration and other system files to guard against system failure.

Perform a Complete System Backup or Restoration

These procedures describe how to back up and restore configurations and images to a tar.gz file and transfer it to your local computer.

Before You Begin Backup or Restore

- You should have at least 300 MB of disk space available at the backup or restore location before you start a backup or restore.
- If you make any configuration changes during or after a backup, those changes will not be included in the backup. If you change a configuration after making the backup, then perform a restore, this configuration change will be overwritten. As a result, the ASA might behave differently.
- You can start only one backup or restore at a time.

- You can only restore a configuration to the same ASA version as when you performed the original backup. You cannot use the restore tool to migrate a configuration from one ASA version to another. If a configuration migration is required, the ASA automatically upgrades the resident startup configuration when it loads the new ASA OS.
- If you use clustering, you can only back up or restore the startup-configuration, running-configuration, and identity certificates. You must create and restore a backup separately for each unit.
- If you use failover, you must create and restore a backup separately for the active and standby units.
- If you set a master passphrase for the ASA, then you need that master passphrase to restore the backup configuration that you create with this procedure. If you do not know the master passphrase for the ASA, see Configure the Master Passphrase, on page 509 to learn how to reset it before continuing with the backup.
- If you import PKCS12 data (with the **crypto ca trustpoint** command) and the trustpoint uses RSA keys, the imported key pair is assigned the same name as the trustpoint. Because of this limitation, if you specify a different name for the trustpoint and its key pair after you have restored an ASDM configuration, the startup configuration will be the same as the original configuration, but the running configuration will include a different key pair name. This means that if you use different names for the key pair and trustpoint, you cannot restore the original configuration. To work around this issue, make sure that you use the same name for the trustpoint and its key pair.
- You cannot back up using the CLI and restore using ASDM, or vice versa.
- Each backup file includes the following content:
  - Running-configuration
  - Startup-configuration
  - All security images

    Cisco Secure Desktop and Host Scan images

    Cisco Secure Desktop and Host Scan settings

    AnyConnect (SVC) client images and profiles

    AnyConnect (SVC) customizations and transforms
  - Identity certificates (includes RSA key pairs tied to identity certificates; excludes standalone keys)
  - VPN pre-shared keys
  - SSL VPN configurations
  - Application Profile Custom Framework (APCF)
  - Bookmarks
  - Customizations
  - Dynamic Access Policy (DAP)
  - Plug-ins
  - Pre-fill scripts for connection profiles
  - Proxy Auto-config

- Translation table
- Web content
- Version information

Back Up the System

This procedure describes how to perform a complete system backup.

Procedure

Step 1 Back up the system:

backup [**/noconfirm**] [**context** *ctx-name*] [**interface** *name*] [**passphrase** *value*] [**location** *path*]

Example:

```
ciscoasa# backup location disk0:/sample-backup]
Backup location [disk0:/sample-backup]?
```

In multiple context mode from the system execution space, enter the **context** keyword to backup the specified context. Each context must be backed up individually; that is, re-enter the **backup** command for each file.

During the backup of VPN certificates and preshared keys, a secret key identified by the **passphrase** keyword is required to encode the certificates. You must provide a passphrase to be used for encoding and decoding the certificates in PKCS12 format. The backup only includes RSA key pairs tied to the certificates and excludes any standalone certificates.

The backup **location** can be a local disk or a remote URL. If you do not provide a location, the following default names are used:

- Single mode—disk0:*hostname*.backup.*timestamp*.tar.gz
- Multiple mode—disk0:*hostname*.context-*ctx-name*.backup.*timestamp*.tar.gz

Step 2 Follow the prompts:

Example:

```
ciscoasa# backup location disk0:/sample-backup
Backup location [disk0:/sample-backup]?

Begin backup…
Backing up [ASA version] … Done!
Backing up [Running Config] … Done!
Backing up [Startup Config]  … Done!

Enter a passphrase to encrypt identity certificates. The default is cisco.
You will be required to enter the same passphrase while doing a restore: cisco
Backing up [Identity Certificates] … Done!

IMPORTANT: This device uses master passphrase encryption. If this backup file
is used to restore to a device with a different master passphrase,
you will need to provide the current master passphrase during restore.
Backing up [VPN Pre-shared keys] … Done!
Backing up [SSL VPN Configurations: Application Profile Custom Framework] … Done!
Backing up [SSL VPN Configurations: Bookmarks]… Done!
Backing up [SSL VPN Configurations: Customization] … Done!
Backing up [SSL VPN Configurations: Dynamic Access Policy] … Done!
Backing up [SSL VPN Configurations: Plug-in] … Done!
Backing up [SSL VPN Configurations: Pre-fill scripts for Connection Profile] … Done!
Backing up [SSL VPN Configurations: Proxy auto-config] … Done!
Backing up [SSL VPN Configurations: Translation table] … Done!
Backing up [SSL VPN Configurations: Web Content] … Done!
Backing up [Anyconnect(SVC) client images and profiles] … Done!
Backing up [Anyconnect(SVC) customizations and transforms] … Done!
Backing up [Cisco Secure Desktop and Host Scan images] … Done!
Backing up [UC-IME tickets] … Done!
Compressing the backup directory ... Done!
Copying Backup ... Done!
Cleaning up ... Done!
Backup finished!
```

Restore the Backup

You can specify configurations and images to restore from a zip tar.gz file on your local computer.

Procedure

Step 1 Restore the system from the backup file.

restore [**/noconfirm**] [**context** *ctx-name*] [**passphrase** *value*] [**location** *path*]

Example:

```
ciscoasa# restore location disk0:/5525-2051.backup.2014-07-09-223$
restore location [disk0:/5525-2051.backup.2014-07-09-223251.tar.gz]?
```

When using the **context** keyword to restore multiple contexts, each backed up context file must be restored individually; that is, re-enter the **restore** command for each file.

Step 2 Follow the prompts:

Example:

```
ciscoasa# restore location disk0:/5525-2051.backup.2014-07-09-223$
restore location [disk0:/5525-2051.backup.2014-07-09-223251.tar.gz]?

Copying Backup file to local disk... Done!
Extracting the backup file ... Done!
Warning: The ASA version of the device is not the same as the backup version,
some configurations might not work after restore!
 Do you want to continue? [confirm] y
Begin restore ...
IMPORTANT: This backup configuration uses master passphrase encryption.
Master passphrase is required to restore running configuration,
startup configuration and VPN pre-shared keys.
Backing up [VPN Pre-shared keys] … Done!
Backing up [SSL VPN Configurations: Application Profile Custom Framework] … Done!
Backing up [SSL VPN Configurations: Bookmarks]… Done!
Backing up [SSL VPN Configurations: Customization] … Done!
Backing up [SSL VPN Configurations: Dynamic Access Policy] … Done!
Backing up [SSL VPN Configurations: Plug-in] … Done!
Backing up [SSL VPN Configurations: Pre-fill scripts for Connection Profile] … Done!
Backing up [SSL VPN Configurations: Proxy auto-config] … Done!
Backing up [SSL VPN Configurations: Translation table] … Done!
Backing up [SSL VPN Configurations: Web Content] … Done!
Backing up [Anyconnect(SVC) client images and profiles] … Done!
Backing up [Anyconnect(SVC) customizations and transforms] … Done!
Backing up [Cisco Secure Desktop and Host Scan images] … Done!
Backing up [UC-IME tickets] … Done!
Restoring [Running Configuration]
Following messages are as a result of applying the backup running-configuration to
this device, please note them for future reference.

ERROR: Interface description was set by failover and cannot be changed
ERROR: Unable to set this url, it has already been set
Remove the first instance before adding this one
INFO: No change to the stateful interface
Failed to update LU link information
.Range already exists.
WARNING: Advanced settings and commands should only be altered or used
under Cisco supervision.
ERROR: Failed to apply media termination address 198.0.1.228 to interface outside,
the IP is already used as media-termination address on interface outside.
ERROR: Failed to apply media termination address 198.0.0.223 to interface inside,
the IP is already used as media-termination address on interface inside.
WARNING: PAC settings will override http- and https-proxy configurations.
Do not overwrite configuration file if you want to preserve the old http-
and https-proxy configurations.

Cryptochecksum (changed): 98d23c2c ccb31dc3 e51acf88 19f04e28
 Done!
Restoring UC-IME ticket ... Done!
Enter the passphrase used while backup to encrypt identity certificates.
The default is cisco. If the passphrase is not correct, certificates will not be restored.

No passphrase was provided for identity certificates.
Using the default value: cisco. If the passphrase is not correct,
certificates will not be restored.
Restoring Certificates ...
Enter the PKCS12 data in base64 representation....
ERROR: A keypair named Main already exists.
INFO: Import PKCS12 operation completed successfully
. Done!
Cleaning up ... Done!
Restore finished!
```

Back up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local flash memory.

Procedure

Back up the configuring using one of the following server types:

- Copy to a TFTP server:

copy [**/noconfirm**] {**startup-config** | **running-config**} **tftp:**//*server*[/*path*]/*dst_filename*

Example:

```
ciscoasa# copy running-config tftp://10.1.1.67/files/new-running.cfg
```

- Copy to an FTP server:

  copy [**/noconfirm**] {**startup-config** | **running-config**} **ftp:**//[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*

  Example:

  ```
  ciscoasa# copy startup-config ftp://jcrichton:aeryn@10.1.1.67/files/new-startup.cfg
  ```

- Copy to an SMB server:

  copy [**/noconfirm**] {**startup-config** | **running-config**}
  smb://[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*

  Example:

  ```
  ciscoasa# copy /noconfirm running-config smb://chiana:dargo@10.1.1.67/new-running.cfg
  ```

- Copy to a SCP server:

  copy [**/noconfirm**] {**startup-config** | **running-config**}
  scp://[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*[**;int=***interface_name*]

  Example:

  ```
  ciscoasa# copy startup-config
  scp://pilot:moya@10.86.94.170/new-startup.cfg
  ```

  The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the SCP server.

- Copy to the local flash memory:

  copy [**/noconfirm**] {**startup-config** | **running-config**} {**disk0**|**disk1**}**:**/[*path*/]*dst_filename*

  Example:

  ```
  ciscoasa# copy /noconfirm running-config disk0:/new-running.cfg
  ```

  Be sure that the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

Back Up a Context Configuration or Other File in Flash Memory

Copy context configurations or other files that are on the local flash memory by entering one of the following commands in the system execution space.

Procedure

Back up a context configuration using one of the following server types:

- Copy from flash to a TFTP server:

  copy [**/noconfirm**] {**disk0**|**disk1**}**:/**[*path*/]*src_filename* **tftp://***server*[/*path*]/*dst_filename*

  Example:

  ```
  ciscoasa# copy disk0:/asa-os.bin tftp://10.1.1.67/files/asa-os.bin
  ```

- Copy from flash to an FTP server:

  copy [**/noconfirm**] {**disk0**|**disk1**}**:/**[*path*/]*src_filename* **ftp://**[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*

  Example:

  ```
  ciscoasa# copy disk0:/asa-os.bin ftp://jcrichton:aeryn@10.1.1.67/files/asa-os.bin
  ```

- Copy from flash to an SMB server:

  copy [**/noconfirm**] {**disk0**|**disk1**}**:/**[*path*/]*src_filename* **smb://**[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*

  Example:

  ```
  ciscoasa# copy /noconfirm copy disk0:/asdm.bin
  smb://chiana:dargo@10.1.1.67/asdm.bin
  ```

- Copy from flash to SCP server:

  copy [**/noconfirm**] {**disk0**|**disk1**}**:/**[*path*/]*src_filename* **scp://**[*user*[**:***password*]**@**]*server*[/*path*]/*dst_filename*[**;int=***interface_name*]

  Example:

  ```
  ciscoasa# copy disk0:/context1.cfg
  scp://pilot:moya@10.86.94.170/context1.cfg
  ```

  The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the SCP server.

- Copy from flash to the local flash memory:

  copy [**/noconfirm**] {**disk0**|**disk1**}**:/**[*path*/]*src_filename* {**disk0**|**disk1**}**:/**[*path*/]*dst_filename*

  Example:

  ```
  ciscoasa# copy /noconfirm disk1:/file1.cfg disk0:/file1.cfgnew-running.cfg
  ```

Be sure that the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

Back Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups.

Procedure

Step 1 Copy the running configuration to the startup configuration server (connected to the admin context):

```
ciscoasa/contexta# copy running-config startup-config
```

Step 2 Copy the running configuration to a TFTP server connected to the context network:

```
ciscoasa/contexta# copy running-config tftp:/server[/path]/filename
```

Copy the Configuration from the Terminal Display

Procedure

Step 1 Print the configuration to the terminal:

more system:running-config

Step 2 Copy the output from this command, and then paste the configuration into a text file.

Back Up Additional Files Using the Export and Import Commands

Additional files essential to your configuration might include the following:

- Files that you import using the **import webvpn** command. Currently, these files include customizations, URL lists, web content, plug-ins, and language translations.
- DAP policies (dap.xml).
- CSD configurations (data.xml).
- Digital keys and certificates.
- Local CA user database and certificate status files.

The CLI lets you back up and restore individual elements of your configuration using the **export** and **import** commands.

To back up these files, for example, those files that you imported with the **import webvpn** command or certificates, perform the following steps.

Procedure

Step 1 Run the applicable **show** command(s) as follows:

```
ciscoasa # show import webvpn plug-in
ica
rdp
ssh, telnet
vnc
```

Step 2 Run the **export** command for the file that you want to back up (in this example, the rdp file):

```
ciscoasa # export webvpn plug-in protocol rdp tftp://tftpserver/backupfilename
```

Use a Script to Back Up and Restore Files

You can use a script to back up and restore the configuration files on your ASA, including all extensions that you import via the **import webvpn** CLI, the CSD configuration XML files, and the DAP configuration XML file. For security reasons, we do not recommend that you perform automated backups of digital keys and certificates or the local CA key.

This section provides instructions for doing so and includes a sample script that you can use as is or modify as your environment requires. The sample script is specific to a Linux system. To use it for a Microsoft Windows system, you need to modify it using the logic of the sample.

Note You can alternatively use the **backup** and **restore** commands. See Perform a Complete System Backup or Restoration, on page 894 for more information.

Before You Begin Using Backup and Restore Scripts

To use a script to back up and restore an ASA configuration, first perform the following tasks:

- Install Perl with an Expect module.
- Install an SSH client that can reach the ASA.
- Install a TFTP server to send files from the ASA to the backup site.

Another option is to use a commercially available tool. You can put the logic of this script into such a tool.

Run the Script

To run a backup-and-restore script, perform the following steps.

Procedure

Step 1 Download or cut-and-paste the script file to any location on your system.

Step 2 At the command line, enter **Perl***scriptname*, where *scriptname* is the name of the script file.

Step 3 Press **Enter**.

Step 4 The system prompts you for values for each option. Alternatively, you can enter values for the options when you enter the **Perl***scriptname* command before you press **Enter**. Either way, the script requires that you enter a value for each option.

Step 5 The script starts running, printing out the commands that it issues, which provides you with a record of the CLIs. You can use these CLIs for a later restore, which is particularly useful if you want to restore only one or two files.

Sample Script

```
#!/usr/bin/perl
#Description: The objective of this script is to show how to back up
configurations/extensions.
# It currently backs up the running configuration, all extensions imported via "import
webvpn" command, the CSD configuration XML file, and the DAP configuration XML file.
#Requirements: Perl with Expect, SSH to the ASA, and a TFTP server.
#Usage: backupasa -option option_value
#       -h: ASA hostname or IP address
#       -u: User name to log in via SSH
#       -w: Password to log in via SSH
#       -e: The Enable password on the security appliance
#       -p: Global configuration mode prompt
#       -s: Host name or IP address of the TFTP server to store the configurations
#       -r: Restore with an argument that specifies the file name. This file is produced
during backup.
#If you don't enter an option, the script will prompt for it prior to backup.
#
#Make sure that you can SSH to the ASA.

use Expect;
use Getopt::Std;

#global variables
%options=();
$restore = 0; #does backup by default
$restore_file = '';
$asa = '';
$storage = '';
$user = '';
$password = '';
$enable = '';
$prompt = '';
$date = `date +%F`;
chop($date);
my $exp = new Expect();
```

```
getopts("h:u:p:w:e:s:r:",\%options);
do process_options();

do login($exp);
do enable($exp);
if ($restore) {
   do restore($exp,$restore_file);
}
else {
   $restore_file = "$prompt-restore-$date.cli";
   open(OUT,">$restore_file") or die "Can't open $restore_file\n";
   do running_config($exp);
   do lang_trans($exp);
   do customization($exp);
   do plugin($exp);
   do url_list($exp);
   do webcontent($exp);
   do dap($exp);
   do csd($exp);
   close(OUT);
}
do finish($exp);

sub enable {
   $obj = shift;
   $obj->send("enable\n");
   unless ($obj->expect(15, 'Password:')) {
      print "timed out waiting for Password:\n";
   }
   $obj->send("$enable\n");
   unless ($obj->expect(15, "$prompt#")) {
      print "timed out waiting for $prompt#\n";
   }
}

sub lang_trans {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("show import webvpn translation-table\n");
   $obj->expect(15, "$prompt#" );
   $output = $obj->before();
   @items = split(/\n+/, $output);

   for (@items) {
     s/^\s+//;
     s/\s+$//;
     next if /show import/ or /Translation Tables/;
     next unless (/^.+\s+.+$/);
     ($lang, $transtable) = split(/\s+/,$_);
     $cli = "export webvpn translation-table $transtable language $lang
$storage/$prompt-$date-$transtable-$lang.po";
     $ocli = $cli;
     $ocli =~ s/^export/import/;
     print "$cli\n";
     print OUT "$ocli\n";
     $obj->send("$cli\n");
     $obj->expect(15, "$prompt#" );
   }
}

sub running_config {
  $obj = shift;
  $obj->clear_accum();
  $cli ="copy /noconfirm running-config $storage/$prompt-$date.cfg";
  print "$cli\n";
  $obj->send("$cli\n");
  $obj->expect(15, "$prompt#" );
}


sub customization {
  $obj = shift;
  $obj->clear_accum();
  $obj->send("show import webvpn customization\n");
  $obj->expect(15, "$prompt#" );
  $output = $obj->before();
  @items = split(/\n+/, $output);

  for (@items) {
    chop;
    next if /^Template/ or /show import/ or /^\s*$/;
    $cli = "export webvpn customization $_ $storage/$prompt-$date-cust-$_.xml";
    $ocli = $cli;
    $ocli =~ s/^export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
  }
}

sub plugin {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("show import webvpn plug-in\n");
   $obj->expect(15, "$prompt#" );
   $output = $obj->before();
   @items = split(/\n+/, $output);

   for (@items) {
     chop;
     next if /^Template/ or /show import/ or /^\s*$/;
     $cli = "export webvpn plug-in protocol $_ $storage/$prompt-$date-plugin-$_.jar";
     $ocli = $cli;
     $ocli =~ s/^export/import/;
     print "$cli\n";
     print OUT "$ocli\n";
     $obj->send("$cli\n");
     $obj->expect(15, "$prompt#" );
   }
}

sub url_list {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("show import webvpn url-list\n");
   $obj->expect(15, "$prompt#" );
   $output = $obj->before();
   @items = split(/\n+/, $output);

   for (@items) {
     chop;
     next if /^Template/ or /show import/ or /^\s*$/ or /No bookmarks/;
     $cli="export webvpn url-list $_ $storage/$prompt-$date-urllist-$_.xml";
     $ocli = $cli;
     $ocli =~ s/^export/import/;
     print "$cli\n";
     print OUT "$ocli\n";
     $obj->send("$cli\n");
     $obj->expect(15, "$prompt#" );
   }
}

sub dap {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("dir dap.xml\n");
   $obj->expect(15, "$prompt#" );

   $output = $obj->before();
   return 0 if($output =~ /Error/);

   $cli="copy /noconfirm dap.xml $storage/$prompt-$date-dap.xml";
   $ocli="copy /noconfirm $storage/$prompt-$date-dap.xml disk0:/dap.xml";
   print "$cli\n";
   print OUT "$ocli\n";
   $obj->send("$cli\n");
   $obj->expect(15, "$prompt#" );
}

sub csd {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("dir sdesktop\n");
   $obj->expect(15, "$prompt#" );

   $output = $obj->before();
   return 0 if($output =~ /Error/);

   $cli="copy /noconfirm sdesktop/data.xml $storage/$prompt-$date-data.xml";
   $ocli="copy /noconfirm $storage/$prompt-$date-data.xml disk0:/sdesktop/data.xml";
   print "$cli\n";
   print OUT "$ocli\n";
   $obj->send("$cli\n");
   $obj->expect(15, "$prompt#" );
}

sub webcontent {
   $obj = shift;
   $obj->clear_accum();
   $obj->send("show import webvpn webcontent\n");
   $obj->expect(15, "$prompt#" );
   $output = $obj->before();
   @items = split(/\n+/, $output);

   for (@items) {
     s/^\s+//;
     s/\s+$//;
     next if /show import/ or /No custom/;
     next unless (/^.+\s+.+$/);
     ($url, $type) = split(/\s+/,$_);
     $turl = $url;
     $turl =~ s/\/\+//;
     $turl =~ s/\+\//-/;
     $cli = "export webvpn webcontent $url $storage/$prompt-$date-$turl";
     $ocli = $cli;
     $ocli =~ s/^export/import/;
     print "$cli\n";
     print OUT "$ocli\n";
     $obj->send("$cli\n");
     $obj->expect(15, "$prompt#" );
   }
```

```
}

sub login {
    $obj = shift;
    $obj->raw_pty(1);
    $obj->log_stdout(0); #turn off console logging.
    $obj->spawn("/usr/bin/ssh $user\@$asa") or die "can't spawn ssh\n";
    unless ($obj->expect(15, "password:" )) {
        die "timeout waiting for password:\n";
    }

    $obj->send("$password\n");

    unless ($obj->expect(15, "$prompt>" )) {
        die "timeout waiting for $prompt>\n";
    }
}

sub finish {
    $obj = shift;
    $obj->hard_close();
    print "\n\n";

}

sub restore {
   $obj = shift;
   my $file = shift;
   my $output;
   open(IN,"$file") or die "can't open $file\n";
   while (<IN>) {
      $obj->send("$_");
      $obj->expect(15, "$prompt#" );
      $output = $obj->before();
      print "$output\n";
   }
   close(IN);
}

sub process_options {
  if (defined($options{s})) {
      $tstr= $options{s};
      $storage = "tftp://$tstr";
  }
  else {
      print "Enter TFTP host name or IP address:";
      chop($tstr=<>);
      $storage = "tftp://$tstr";
  }
  if (defined($options{h})) {
      $asa = $options{h};
  }
  else {
      print "Enter ASA host name or IP address:";
      chop($asa=<>);
  }

  if (defined ($options{u})) {
      $user= $options{u};
  }
  else {
      print "Enter user name:";
      chop($user=<>);
  }
  if (defined ($options{w})) {
      $password= $options{w};
  }
  else {
      print "Enter password:";
      chop($password=<>);
  }
  if (defined ($options{p})) {
      $prompt= $options{p};
  }
  else {
      print "Enter ASA prompt:";
      chop($prompt=<>);
  }
  if (defined ($options{e})) {
      $enable = $options{e};
  }
  else {
      print "Enter enable password:";
      chop($enable=<>);
  }

  if (defined ($options{r})) {
     $restore = 1;
     $restore_file = $options{r};
  }
}
```

Configure Auto Update

Auto Update is a protocol specification that allows an Auto Update Server to download configurations and software images to many ASAs and can provide basic monitoring of the ASAs from a central location.

About Auto Update

This section describes how Auto Update is implemented and why you might want to use Auto Update.

Auto Update Client or Server

The ASA can be configured as either a client or a server. As an Auto Update client, it periodically polls the Auto Update Server for updates to software images and configuration files. As an Auto Update Server, it issues updates for ASAs configured as Auto Update clients.

Auto Update Benefits

Auto Update is useful in solving many issues facing administrators for ASA management, such as:

- Overcoming dynamic addressing and NAT challenges.
- Committing configuration changes in one action.
- Providing a reliable method for updating software.
- Leveraging well-understood methods for high availability (failover).

- Providing flexibility with an open interface.
- Simplifying security solutions for Service Provider environments.

The Auto Update specification provides the infrastructure necessary for remote management applications to download ASA configurations, software images, and to perform basic monitoring from a centralized location or multiple locations.

The Auto Update specification allows the Auto Update server to either push configuration information and send requests for information to the ASA, or to pull configuration information by having the ASA periodically poll the Auto Update server. The Auto Update server can also send a command to the ASA to send an immediate polling request at any time. Communication between the Auto Update server and the ASA requires a communications path and local CLI configuration on each ASA.

Auto Update Server Support in Failover Configurations

You can use the Auto Update Server to deploy software images and configuration files to ASAs in an Active/Standby failover configuration. To enable Auto Update on an Active/Standby failover configuration, enter the Auto Update Server configuration on the primary unit in the failover pair.

The following restrictions and behaviors apply to Auto Update Server support in failover configurations:

- Only single mode, Active/Standby configurations are supported.
- When loading a new platform software image, the failover pair stops passing traffic.
- When using LAN-based failover, new configurations must not change the failover link configuration. If they do, communication between the units will fail.
- Only the primary unit will perform the call home to the Auto Update Server. The primary unit must be in the active state to call home. If it is not, the ASA automatically fails over to the primary unit.
- Only the primary unit downloads the software image or configuration file. The software image or configuration is then copied to the secondary unit.
- The interface MAC address and hardware-serial ID is from the primary unit.
- The configuration file stored on the Auto Update Server or HTTP server is for the primary unit only.

Auto Update Process Overview

The following is an overview of the Auto Update process in failover configurations. This process assumes that failover is enabled and operational. The Auto Update process cannot occur if the units are synchronizing configurations, if the standby unit is in the failed state for any reason other than SSM card failure, or if the failover link is down.

1. Both units exchange the platform and ASDM software checksum and version information.
2. The primary unit contacts the Auto Update Server. If the primary unit is not in the active state, the ASA first fails over to the primary unit and then contacts the Auto Update Server.
3. The Auto Update Server replies with software checksum and URL information.
4. If the primary unit determines that the platform image file needs to be updated for either the active or standby unit, the following occurs:
    a. The primary unit retrieves the appropriate files from the HTTP server using the URL from the Auto Update Server.

b. The primary unit copies the image to the standby unit and then updates the image on itself.

c. If both units have new image, the secondary (standby) unit is reloaded first.

   - If hitless upgrade can be performed when secondary unit boots, then the secondary unit becomes the active unit and the primary unit reloads. The primary unit becomes the active unit when it has finished loading.

   - If hitless upgrade cannot be performed when the standby unit boots, then both units reload at the same time.

d. If only the secondary (standby) unit has new image, then only the secondary unit reloads. The primary unit waits until the secondary unit finishes reloading.

e. If only the primary (active) unit has new image, the secondary unit becomes the active unit, and the primary unit reloads.

f. The update process starts again at Step 1.

5. If the ASA determines that the ASDM file needs to be updated for either the primary or secondary unit, the following occurs:

   a. The primary unit retrieves the ASDM image file from the HTTP server using the URL provided by the Auto Update Server.

   b. The primary unit copies the ASDM image to the standby unit, if needed.

   c. The primary unit updates the ASDM image on itself.

   d. The update process starts again at Step 1.

6. If the primary unit determines that the configuration needs to be updated, the following occurs:

   a. The primary unit retrieves the configuration file from the using the specified URL.

   b. The new configuration replaces the old configuration on both units simultaneously.

   c. The update process begins again at Step 1.

7. If the checksums match for all image and configuration files, no updates are required. The process ends until the next poll time.

Guidelines for Auto Update

Context Mode

Auto Update is supported in single context mode only.

Clustering

No clustering support.

Models

No support on the following models:

- ASA 5506-X, 5508-X, 5516-X
- Firepower 4100, and 9300
- ASAv

Additional Guidelines

- If HTTPS is chosen as the protocol to communicate with the Auto Update server, the ASA uses SSL, which requires the ASA to have a DES or 3DES license.

Configure Communication with an Auto Update Server

Procedure

Step 1 To specify the URL of the Auto Update Server, enter the following command:

auto-update server *url* [**source** *interface*] [**verify-certificate** | **no-verification**]

where *url* has the following syntax:

http[**s**]**://**[*user***:***password***@**]*server_ip*[**:***port*]**/***pathname*

The **source** *interface* keyword and argument specify which interface to use when sending requests to the Auto Update Server. If you specify the same interface specified by the **management-access** command, the Auto Update requests travel over the same IPsec VPN tunnel used for management access.

For HTTPS, the **verify-certificate** keyword (the default) verifies the certificate returned by the Auto Update Server. To disable verification (not recommended), specify the **no-verification** keyword.

Step 2 (Optional) To identify the device ID to send when communicating with the Auto Update Server, enter the following command:

auto-update device-id {*hardware-serial* | **hostname** | **ipaddress** [*if-name*] | **mac-address** [*if-name*] | **string** *text*}

The identifier used is determined by specifying one of the following parameters:

- The hardware-serial argument specifies the ASA serial number.
- The hostname argument specifies the ASA hostname.
- The **ipaddress** keyword specifies the IP address of the specified interface. If the interface name is not specified, it uses the IP address of the interface used to communicate with the Auto Update Server.
- The **mac-address** keyword specifies the MAC address of the specified interface. If the interface name is not specified, it uses the MAC address of the interface used to communicate with the Auto Update Server.
- The **string** keyword specifies the specified text identifier, which cannot include white space or the characters ‘, “, , >, & and ?.

Step 3 (Optional) To specify how often to poll the Auto Update Server for configuration or image updates, enter the following command:

auto-update poll-period *poll-period* [*retry-count* [*retry-period*]]

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is zero.

The *retry-period* argument specifies how long to wait (in minutes) between retries. The default is five minutes.

Step 4 (Optional) To schedule a specific time for the ASA to poll the Auto Update Server, enter the following command:

auto-update poll-at *days-of-the-week time* [**randomize** *minutes*] [*retry_count* [*retry_period*]]

The *days-of-the-week* argument is any single day or combination of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Other possible values are daily (Monday through Sunday), weekdays (Monday through Friday), and weekends (Saturday and Sunday).

The *time* argument specifies the time in the format HH:MM at which to start the poll. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

The **randomize** *minutes* keyword and argument specify the period to randomize the poll time following the specified start time. The range is from 1 to 1439 minutes.

The *retry_count* argument specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is zero.

The *retry_period* argument specifies how long to wait between connection attempts. The default is five minutes. The range is from 1 to 35791 minutes.

Step 5 (Optional) If the Auto Update Server has not been contacted for a certain period of time, entering the following command causes it to stop passing traffic:

auto-update timeout *period*

The period argument specifies the timeout period in minutes between 1 and 35791. The default is to never time out (zero minutes). To restore the default, enter the **no** form of this command.

Use the **auto-update timeout** command to be sure that the ASA has the most recent image and configuration. This condition is reported with system log message 201008.

Example

In the following example, an ASA is configured to poll an Auto Update Server with the IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

The ASA is also configured to use the hostname as the device ID and to poll an Auto Update Server every Friday and Saturday night at a random time between 10:00 p.m. and 11:00 p.m. On a failed polling attempt, the ASA will try to reconnect to the Auto Update Server ten times, and will wait three minutes between attempts at reconnecting, as shown in the following example:

```
ciscoasa(config)# auto-update server
https://jcrichton:farscape@209.165.200.224:1742/management source outside verify-certificate
ciscoasa (config)# auto-update device-id hostname
hostname (config)# auto-update poll-at Friday Saturday 22:00 randomize 60 2 10
```

Configure Client Updates as an Auto Update Server

Entering the **client-update** command enables updates for ASAs configured as Auto Update clients and lets you specify the type of software component (ASDM or boot image), the type or family of ASA, revision numbers to which the update applies, and a URL or IP address from which to obtain the update.

To configure the ASA as an Auto Update Server, perform the following steps.

Procedure

Step 1 To enable client update, enter the following command:

```
ciscoasa(config)# client-update enable
```

Step 2 Configure the following parameters for the **client-update** command that you want to apply to the ASAs:

client-update {**component** {**asdm** | **image**} | **device-id** dev_string | **family** family_name | **type** type} **url** url-string **rev-nums** rev-nums}

The **component** {**asdm** | **image**} parameter specifies the software component, either ASDM or the boot image of the ASA.

The **device-id** dev_string parameter specifies a unique string that the Auto Update client uses to identify itself. The maximum length is 63 characters.

The **family** family_name parameter specifies the family name that the Auto Update client uses to identify itself. It can be asa, pix, or a text string with a maximum length of seven characters.

The **rev-nums** rev-nums parameter specifies the software or firmware images for this client. Enter up to four, in any order, separated by commas.

The **type** *type* parameter specifies the type of clients to notify of a client update. Because this command is also used to update Windows clients, the list of clients includes several Windows operating systems.

The **url** url-string parameter specifies the URL for the software/firmware image. This URL must point to a file appropriate for this client. For all Auto Update clients, you must use the protocol “http://” or “https://” as the prefix for the URL.

Configure the parameters for the client update that you want to apply to all ASAs of a particular type. That is, specify the type of ASA and the URL or IP address from which to get the updated image. In addition, you must specify a revision number. If the revision number of the remote ASA matches one of the specified revision numbers, there is no need to update the client, and the update is ignored.

To configure a client update for ASA 5525-Xs, enter the following command:

```
ciscoasa(config)# client-update type asa5525 component asdm url
http://192.168.1.114/aus/asdm710.bin rev-nums 9.10(1)
```

Monitoring Auto Update

Monitoring the Auto Update Process

You can use the **debug auto-update client** or **debug fover cmd-exe** commands to display the actions performed during the Auto Update process. The following is sample output from the **debug auto-update client** command.

```
Auto-update client: Sent DeviceDetails to /cgi-bin/dda.pl of server 192.168.0.21
Auto-update client: Processing UpdateInfo from server 192.168.0.21
   Component: asdm, URL: http://192.168.0.21/asdm.bint, checksum:
0x94bced0261cc992ae710faf8d244cf32
   Component: config, URL: http://192.168.0.21/config-rms.xml, checksum:
0x67358553572688a805a155af312f6898
   Component: image, URL: http://192.168.0.21/cdisk73.bin, checksum:
0x6d091b43ce96243e29a62f2330139419
Auto-update client: need to update img, act: yes, stby yes
name
ciscoasa(config)# Auto-update client: update img on stby unit...
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 9001, len = 1024
auto-update: Fover file copy waiting at clock tick 6129280
fover_parse: Rcvd file copy ack, ret = 0, seq = 4
auto-update: Fover filecopy returns value: 0 at clock tick 6150260, upd time 145980 msecs
Auto-update client: update img on active unit...
fover_parse: Rcvd image info from mate
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
Beginning configuration replication: Sending to mate.
auto-update: HA safe reload: reload active waiting with mate state: 50
auto-update: HA safe reload: reload active waiting with mate state: 50
```

```
auto-update: HA safe reload: reload active waiting with mate state: 80
        Sauto-update: HA safe reload: reload active unit at clock tick: 6266860
Auto-update client: Succeeded: Image, version: 0x6d091b43ce96243e29a62f2330139419
```

The following syslog message is generated if the Auto Update process fails:

```
%ASA4-612002: Auto Update failed: file version: version reason: reason
```

The *file* is "image", "asdm", or "configuration", depending on which update failed. The *version* is the version number of the update. And the *reason* is the reason that the update failed.

Monitoring Auto Update Status

See the following command for monitoring Auto Update status:

show auto-update

The following is sample output from the **show auto-update** command:

```
ciscoasa(config)# show auto-update

Server: https://********@209.165.200.224:1742/management.cgi?1276
Certificate will be verified
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes
Timeout: none
Device ID: host name [corporate]
Next poll in 4.93 minutes
Last poll: 11:36:46 PST Tue Nov 13 2004
Last PDM update: 23:36:46 PST Tue Nov 12 2004
```

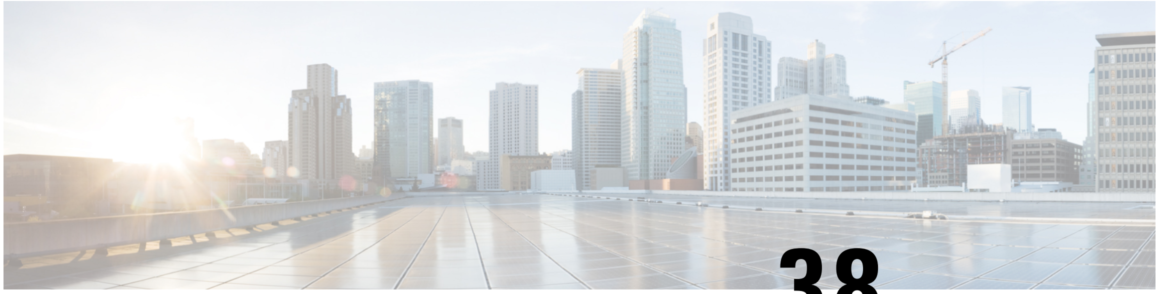
History for Software and Configurations

Feature Name	Platform Releases	Feature Information
Secure Copy client	9.1(5)/9.2(1)	The ASA now supports the Secure Copy (SCP) client to transfer files to and from a SCP server. We introduced the following commands: **ssh pubkey-chain**, **server (ssh pubkey-chain)**, **key-string**, **key-hash**, ssh **stricthostkeycheck**. We modified the following command: **copy scp**.

Feature Name	Platform Releases	Feature Information
Configurable SSH encryption and integrity ciphers	9.1(7)/9.4(3)	Users can select cipher modes when doing SSH encryption management and can configure HMAC and encryption for varying key exchange algorithms. You might want to change the ciphers to be more or less strict, depending on your application. Note that the performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use **ssh cipher encryption custom aes128-cbc**, for example. We introduced the following commands: **ssh cipher encryption**, **ssh cipher integrity**

Auto Update server certificate verification enabled by default	9.2(1)	The Auto Update server certificate verification is now enabled by default; for new configurations, you must explicitly disable certificate verification. If you are upgrading from an earlier release, and you did not enable certificate verification, then certificate verification is not enabled, and you see the following warning:
		`WARNING: The certificate provided by the`
		`auto-update servers will not be verified.`
		`In order to verify this certificate please`
		`use the verify-certificate option.`
		The configuration will be migrated to explicitly configure no verification.
		auto-update server no-verification
		We modified the following command: **auto-update server** {**verify-certificate** | **no-verification**}.

System backup and restore using the CLI	9.3(2)	You can now back up and restore complete system configurations, including images and certificates, using the CLI. We introduced the following commands: **backup** and **restore**.
Recovering and loading a new ASA 5506W-X image	9.4(1)	We now support the recovery and loading of a new ASA 5506W-X image. We introduced the following command: **hw-module module wlan recover image**.



CHAPTER 38

Response Automation for System Events

This chapter describes how to configure the Embedded Event Manager (EEM).

About the EEM

The EEM service enables you to debug problems and provides general purpose logging for troubleshooting. There are two components: events to which the EEM responds or listens, and event manager applets that define actions as well as the events to which the EEM responds. You may configure multiple event manager applets to respond to different events and perform different actions.

Supported Events

The EEM supports the following events:

- Syslog—The ASA uses syslog message IDs to identify syslog messages that trigger an event manager applet. You may configure multiple syslog events, but the syslog message IDs may not overlap within a single event manager applet.
- Timers—You may use timers to trigger events. You may configure each timer only once for each event manager applet. Each event manager applet may have up to three timers. The three types of timers are the following:
  - Watchdog (periodic) timers trigger an event manager applet after the specified time period following the completion of the applet actions and restart automatically.
  - Countdown (one-shot) timers trigger an event manager applet once after the specified time period and do not restart unless they are removed, then re-added.
  - Absolute (once-a-day) timers cause an event to occur once a day at a specified time, and restart automatically. The time-of-day format is in hh:mm:ss.

    You may configure only one timer event of each type for each event manager applet.

- None—The none event is triggered when you run an event manager applet manually using the CLI or ASDM.
- Crash—The crash event is triggered when the ASA crashes. Regardless of the value of the **output** command, the **action** commands are directed to the crashinfo file. The output is generated before the **show tech** command.

Actions on Event Manager Applets

When an event manager applet is triggered, the actions on the event manager applet are performed. Each action has a number that is used to specify the sequence of the actions. The sequence number must be unique within an event manager applet. You may configure multiple actions for an event manager applet. The commands are typical CLI commands, such as **show blocks**.

Output Destinations

You may send the output from the actions to a specified location using the **output** command. Only one output value may be enabled at any one time. The default value is **output none**. This value discards any output from the **action** commands. The command runs in global configuration mode as a user with privilege level 15 (the highest). The command may not accept any input, because it is disabled. You may send the output of the **action** CLI commands to one of three locations:

- **None**, which is the default and discards the output
- **Console**, which sends the output to the ASA console
- **File**, which sends the output to a file. The following four file options are available:
  - **Create a unique file**, which creates a new, uniquely named file each time that an event manager applet is invoked
  - **Create/overwrite a file**, which overwrites a specified file each time that an event manager applet is invoked.
  - **Create/append to a file**, which appends to a specified file each time that an event manager applet is invoked. If the file does not yet exist, it is created.
  - **Create a set of files**, which creates a set of uniquely named files that are rotated each time that an event manager applet is invoked.

Guidelines for the EEM

This section describes guidelines and limitations that you should check before configuring the EEM.

Context Mode Guidelines

Not supported in multiple context mode.

Additional Guidelines

- During a crash, the state of the ASA is generally unknown. Some commands may not be safe to run during this condition.
- The name of an event manager applet may not contain spaces.
- You cannot modify the None event and Crashinfo event parameters.
- Performance may be affected because syslog messages are sent to the EEM for processing.
- The default output is **output none** for each event manager applet. To change this setting, you must enter a different output value.
- You may have only one output option defined for each event manager applet.

Configure the EEM

Configuring the EEM consists of the following tasks:

Procedure

Step 1 Create an Event Manager Applet and Configure Events, on page 919.

Step 2 Configure an Action and Destinations for Output from an Action, on page 921.

Step 3 Run an Event Manager Applet, on page 923.

Step 4 Track Memory Allocation and Memory Usage, on page 923.

Create an Event Manager Applet and Configure Events

To create an event manager applet and configure events, perform the following steps:

Procedure

Step 1 Create an event manager applet and enter event manager applet configuration mode.

event manager applet *name*

Example:

```
ciscoasa(config)# event manager applet exampleapplet1
```

The *name* argument may be up to 32 alphanumeric characters long. Spaces are not allowed.

To remove an event manager applet, enter the **no** form of this command.

Step 2 Describe an event manager applet.

description *text*

Example:

```
ciscoasa(config-applet)# description applet1example
```

The *text* argument may be up to 256 characters long. You may include spaces in description text if it is placed within quotes.

Step 3 To configure a specified event, enter one of the following commands. To remove the configured event, enter the **no** form of each of the commands.

- To configure a syslog event, identify a single syslog message or a range of syslog messages that trigger an event manager applet.

  event syslog id *nnnnnn* [-*nnnnnn*] [**occurs** *n*] [**period** *seconds*]

  Example:

  ```
  ciscoasa(config-applet)# event syslog id 106201
  ```

  The *nnnnnn* argument identifies the syslog message ID. The **occurs** *n* keyword-argument pair indicates the number of times that the syslog message must occur for an event manager applet to be invoked. The default is 1 occurrence every 0 seconds. Valid values are from 1 - 4294967295. The **period** *seconds* keyword-argument pair indicates the number of seconds in which the event must occur, and limits how frequently an event manager applet is invoked to at most once in the configured period. Valid values are from 0 - 604800. A value of 0 means that no period is defined.

- To configure an event to occur once per configured period and restart automatically.

  event timer watchdog time *seconds*

  Example:

  ```
  ciscoasa(config-applet)# event timer watchdog time 30
  ```

  The number of seconds may range from 1 - 604800.

- To configure an event to occur once and not restart unless it is removed, then re-added.

  event timer countdown time *seconds*

  Example:

  ```
  ciscoasa(config-applet)# event timer countdown time 60
  ```

  The number of seconds may range from 1 - 604800. Use the **no** form of this command remove a countdown timer event.

  Note This timer reruns when you reboot if it is the startup configuration.

- To configure an event to occur once a day at a specified time and restart automatically.

  event timer absolute time *hh:mm:ss*

  Example:

```
ciscoasa(config-applet)# event timer absolute time 10:30:20
```

The time-of-day format is in hh:mm:ss. The time range is from 00:00:00 (midnight) to 23:59:59.

- Trigger a crash event when the ASA crashes.

  event crashinfo

  Example:

  ```
  ciscoasa(config-applet)# event crashinfo
  ```

  Regardless of the value of the **output** command, the **action** commands are directed to the crashinfo file. The output is generated before the **show tech** command.

Configure an Action and Destinations for Output from an Action

To configure an action and specific destinations for sending output from an action, perform the following steps:

Procedure

Step 1 Configure an action on an event manager applet.

action *n* **cli command** *"command"*

Example:

```
ciscoasa(config-applet)# action 1 cli command "show version"
```

The *n* option is an action ID. Valid IDs range from 0 - 4294967295. The value of the *command* option must be in quotes; otherwise, an error occurs if the command consists of more than one word. The command runs in global configuration mode as a user with privilege level 15 (the highest). The command may not accept any input, because it is disabled. Use the **noconfirm** option if the command has it available.

Step 2 Choose one of the available output destination options. Use the **no** form of each command to remove an output destination,

- The **None** option discards any output from the **action** commands, which is the default setting:

  output none

  Example:

  ```
  ciscoasa(config-applet)# output none
  ```

- The **Console** option sends the output of the **action** commands to the console.

  output console

Example:

```
ciscoasa(config-applet)# output console
```

Note Running this command affects performance.

- The **New File** option sends the output of the **action** commands to a new file for each event manager applet that is invoked.

  output file new

  Example:

  ```
  ciscoasa(config-applet)# output file new
  ```

  The filename has the format of eem-*applet*-*timestamp*.log, in which *applet* is the name of the event manager applet and *timestamp* is a dated time stamp in the format of YYYYMMDD-hhmmss.

- The **New Set of Rotated Files** option creates a set of files that are rotated. When a new file is to be written, the oldest file is deleted, and all subsequent files are renumbered before the first file is written.

  output file rotate *n*

  Example:

  ```
  ciscoasa(config-applet)# output file rotate 50
  ```

  The newest file is indicated by 0, and the oldest file is indicated by the highest number (*n*-1). The *n* option is the rotate value. Valid values range from 2 - 100. The filename format is eem-*applet*-*x*.log, in which *applet* is the name of the applet, and *x* is the file number.

- The **Single Overwritten File** option writes the **action** command output to a single file, which is overwritten every time.

  output file overwrite *filename*

  Example:

  ```
  ciscoasa(config-applet)# output file overwrite examplefile1
  ```

  The *filename* argument is a local (to the ASA) filename. This command may also use FTP, TFTP, and SMB targeted files.

- The **Single Appended File** option writes the **action** command output to a single file, but that file is appended to every time.

  output file append *filename*

  Example:

  ```
  ciscoasa(config-applet)# output file append examplefile1
  ```

The *filename* argument is a local (to the ASA) filename.

Run an Event Manager Applet

To run an event manager applet, perform the following steps:

Procedure

Run an event manager applet.

event manager run *applet*

Example:

```
ciscoasa# event manager run exampleapplet1
```

If you run an event manager applet that has not been configured with the **event none** command, an error occurs. The *applet* argument is the name of the event manager applet.

Track Memory Allocation and Memory Usage

To log memory allocation and memory usage, perform the following steps:

Procedure

Step 1 Enable memory logging.

memory logging [1024-4194304] [**wrap**] [**size** [1-2147483647]] [**process** *process-name*] [**context** *context-name*]

Example:

```
ciscoasa(config)# memory logging 202980
```

The only required argument is the number of entries in the memory logging buffer. The **wrap** option tells the memory logging utility to save the buffer when it wraps. It can only be saved once.

If the memory logging buffer wraps multiple times, it can be overwritten. When the buffer wraps, a trigger is sent to the event manager to enable saving of the data. The **size** option monitors a particular size. The **process** option monitors a particular process.

Note The Checkheaps process is completely ignored as a process because it uses the memory allocator in a non-standard way.

The **context** option records memory logging for a given virtual context by the given name.

To change memory logging parameters, you must disable it, then reenable it.

Step 2 Display the memory logging results.

```
show memory logging [brief | wrap]
show memory logging include [address] [caller] [operator] [size] [process] [time] [context]
```

Example:

```
ciscoasa# show memory logging
Number of free                           6
Number of calloc                         0
Number of malloc                         8
Number of realloc-new                    0
Number of realloc-free                   0
Number of realloc-null                   0
Number of realloc-same                   0
Number of calloc-fail                    0
Number of malloc-fail                    0
Number of realloc-fail                   0
Total operations 14
Buffer size: 50 (3688 x2 bytes)
process=[ci/console] time=[13:26:33.407] oper=[malloc]
addr=0x00007fff2cd0a6c0 size=72 @ 0x00000000016466ea 0x0000000002124542
0x000000000131911a 0x0000000000442bfd process=[ci/console] time=[13:26:33.407] oper=[free]
addr=0x00007fff2cd0a6c0 size=72 @ 0x00000000021246ef 0x00000000013193e8
0x0000000000443455 0x0000000001318f5b
process=[CMGR Server Process] time=[13:26:35.964] oper=[malloc]
addr=0x00007fff2cd0aa00 size=16 @ 0x00000000016466ea 0x0000000002124542
0x000000000182774d 0x000000000182cc8a process=[CMGR Server Process]
time=[13:26:35.964] oper=[malloc]
addr=0x00007fff224bb9f0 size=512 @ 0x00000000016466ea 0x0000000002124542
0x0000000000bfef9a 0x0000000000bff606 process=[CMGR Server Process]
time=[13:26:35.964] oper=[free]
addr=0x00007fff224bb9f0 size=512 @ 0x00000000021246ef 0x0000000000bff3d8
0x0000000000bff606 0x000000000182ccb0
process=[CMGR Server Process] time=[13:26:35.964] oper=[malloc]
addr=0x00007fff224b9460 size=40 @ 0x00000000016466ea 0x0000000002124542
0x0000000001834188 0x000000000182ce83
process=[CMGR Server Process] time=[13:26:37.964] oper=[free]
addr=0x00007fff2cd0aa00 size=16 @ 0x00000000021246ef 0x0000000001827098
0x000000000182c08d 0x000000000182c262 process=[CMGR Server Process]
time=[13:26:37.964] oper=[free]
addr=0x00007fff224b9460 size=40 @ 0x00000000021246ef 0x000000000182711b
0x000000000182c08d 0x000000000182c262 process=[CMGR Server Process]
time=[13:26:38.464] oper=[malloc]
addr=0x00007fff2cd0aa00 size=16 @ 0x00000000016466ea 0x0000000002124542
0x000000000182774d 0x000000000182cc8a process=[CMGR Server Process]
time=[13:26:38.464] oper=[malloc]
addr=0x00007fff224bb9f0 size=512 @ 0x00000000016466ea 0x0000000002124542
0x0000000000bfef9a 0x0000000000bff606 process=[CMGR Server Process]
time=[13:26:38.464] oper=[free]
addr=0x00007fff224bb9f0 size=512 @ 0x00000000021246ef 0x0000000000bff3d8
0x0000000000bff606 0x000000000182ccb0
process=[CMGR Server Process] time=[13:26:38.464] oper=[malloc]
addr=0x00007fff224b9460 size=40 @ 0x00000000016466ea 0x0000000002124542
0x0000000001834188 0x000000000182ce83
process=[ci/console] time=[13:26:38.557] oper=[malloc]
addr=0x00007fff2cd0a6c0 size=72 @ 0x00000000016466ea 0x0000000002124542
0x000000000131911a 0x0000000000442bfd process=[ci/console] time=[13:26:38.557] oper=[free]
addr=0x00007fff2cd0a6c0 size=72 @ 0x00000000021246ef 0x00000000013193e8
```

```
0x0000000000443455 0x0000000001318f5b

ciscoasa# show memory logging include process operation size
Number of free                     6
Number of calloc                   0
Number of malloc                   8
Number of realloc-new              0
Number of realloc-free             0
Number of realloc-null             0
Number of realloc-same             0
Number of calloc-fail              0
Number of malloc-fail              0
Number of realloc-fail             0
Total operations 14
Buffer size: 50 (3688 x2 bytes)
process=[ci/console] oper=[malloc] size=72 process=[ci/console] oper=[free]
size=72 process=[CMGR Server Process] oper=[malloc] size=16
process=[CMGR Server Process] oper=[malloc] size=512 process=[CMGR Server Process]
oper=[free] size=512 process=[CMGR Server Process] oper=[malloc] size=40
process=[CMGR Server Process] oper=[free] size=16 process=[CMGR Server Process]
oper=[free] size=40 process=[CMGR Server Process] oper=[malloc] size=16
process=[CMGR Server Process] oper=[malloc] size=512 process=[CMGR Server Process]
oper=[free] size=512 process=[CMGR Server Process] oper=[malloc] size=40
process=[ci/console] oper=[malloc] size=72 process=[ci/console]
oper=[free] size=72 ciscoasa# show memory logging brief
Number of free                     6
Number of calloc                   0
Number of malloc                   8
Number of realloc-new              0
Number of realloc-free             0
Number of realloc-null             0
Number of realloc-same             0
Number of calloc-fail              0
Number of malloc-fail              0
Number of realloc-fail             0
Total operations 14
Buffer size: 50 (3688 x2 bytes)
```

Without any options, **show memory logging** displays statistics and then the recorded operations. The **brief** option shows only statistics. The **wrap** option shows the buffer upon wrap, then purges the data so that duplicate data does not appear or get saved. The **include** option includes only the specified fields in the output. You can specify the fields in any order, but they always appear in the following order:

a. Process

b. Time

c. Context (unless in single mode)

d. Operation (free/malloc/etc.)

e. Address

f. Size

g. Callers

The output format is:

process=[XXX] time=[XXX] context=[XXX] oper=[XXX] address=0xXXXXXXXX size=XX @ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX

Up to four caller addresses appear. The types of operations are listed in the output (Number of...) shown in the example.

Step 3 Respond to memory logging wrap events.

event memory-logging-wrap

Example:

```
ciscoasa(config)# event manager applet memlog
ciscoasa(config)# event memory-logging-wrap
ciscoasa(config)# action 0 cli command "show memory logging wrap"
ciscoasa(config)# output file append disk0:/memlog.log
```

The example shows an applet that records all memory allocations. When wrap is enabled for memory logging, the memory logger sends an event to the event manager to trigger configured applets.

Examples for the EEM

The following example shows an event manager applet that records block leak information every hour and writes the output to a rotating set of log files, keeping a day's worth of logs:

```
ciscoasa(config)# event manager applet blockcheck
ciscoasa(config-applet)# description “Log block usage”
ciscoasa(config-applet)# event timer watchdog time 3600
ciscoasa(config-applet)# output rotate 24
ciscoasa(config-applet)# action 1 cli command “show blocks old”
```

The following example shows an event manager applet that reboots the ASA every day at 1 am, saving the configuration as needed:

```
ciscoasa(config)# event manager applet dailyreboot
ciscoasa(config-applet)# description “Reboot every night”
ciscoasa(config-applet)# event timer absolute time 1:00:00
ciscoasa(config-applet)# output none
ciscoasa(config-applet)# action 1 cli command “reload save-config noconfirm”
```

The following example shows event manager applets that disable the given interface between midnight and 3 am.

```
ciscoasa(config)# event manager applet disableintf
ciscoasa(config-applet)# description “Disable the interface at midnight”
ciscoasa(config-applet)# event timer absolute time 0:00:00
ciscoasa(config-applet)# output none
ciscoasa(config-applet)# action 1 cli command “interface GigabitEthernet 0/0”
ciscoasa(config-applet)# action 2 cli command “shutdown”
ciscoasa(config-applet)# action 3 cli command “write memory”

ciscoasa(config)# event manager applet enableintf
ciscoasa(config-applet)# description “Enable the interface at 3am”
ciscoasa(config-applet)# event timer absolute time 3:00:00
ciscoasa(config-applet)# output none
ciscoasa(config-applet)# action 1 cli command "interface GigabitEthernet 0/0"
ciscoasa(config-applet)# action 2 cli command "no shutdown"
ciscoasa(config-applet)# action 3 cli command "write memory"
```

Monitoring the EEM

See the following commands to monitor the EEM.

- **clear configure event manager**

  This command removes the event manager running configuration.

- **clear configure event manager applet** *appletname*

  This command removes the named event manager applet from the configuration.

- **show counters protocol eem**

  This command shows the counters for the event manager.

- **show event manager**

  This command shows information about the configured event manager applets, including hit counts and when the event manager applets were last invoked.

- **show memory logging, show memory logging include**

  These commands show statistics about the memory allocations and memory usage.

- **show running-config event manager**

  This command shows the running configuration of the event manager.

History for the EEM

Table 46: History for the EEM

Feature Name	Platform Releases	Description
Embedded Event Manager (EEM)	9.2(1)	The EEM service enables you to debug problems and provides general purpose logging for troubleshooting. There are two components: events to which the EEM responds or listens, and event manager applets that define actions as well as the events to which the EEM responds. You may configure multiple event manager applets to respond to different events and perform different actions. We introduced or modified the following commands: **event manager applet**, **description**, **event syslog id**, **event none**, **event timer** {**watchdog time** *seconds* | **countdown time** *seconds* | **absolute time** *hh:mm:ss*}, **event crashinfo**, **action cli command**, **output** {**none** | **console** | **file** {**append** *filename* | **new** | **overwrite** *filename* | **rotate** *n*}}, **show running-config event manager**, **event manager run**, **show event manager**, **show counters protocol eem**, **clear configure event manager**, **debug event manager**, **debug menu eem**.
Memory tracking for the EEM	9.4(1)	We have added a new debugging feature to log memory allocations and memory usage, and to respond to memory logging wrap events. We introduced or modified the following commands: **memory logging**, **show memory logging**, **show memory logging include**, **event memory-logging-wrap**.

CHAPTER 39

Testing and Troubleshooting

This chapter describes how to troubleshoot the Cisco ASA and test basic connectivity.

Recover Enable and Telnet Passwords

If you forget the enable or Telnet passwords, you can recover them for ASA models. The procedure differs by device type. You must perform the task using the CLI.

Note For Firepower platforms, you cannot recover lost passwords. You can only restore the factory default configuration, and reset the passwords to the default. For Firepower 9300, see the FXOS configuration guide.

Recover Passwords on the ASA 5500-X

This procedure works for the ASA 5512-X, 5515-X, 5525-X, 5545-X, 5555-X, and 5585-X.

To recover passwords for the ASA, perform the following steps.

Procedure

Step 1 Connect to the ASA console port.

Step 2 Power off the ASA, then power it on.

Step 3 After startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 To update the configuration register value, enter the following command:

```
rommon #1> confreg 0x41
Update Config Register (0x41) in NVRAM...
```

Step 5 To set the ASA to ignore the startup configuration, enter the following command:

```
rommon #1> confreg
```

The ASA displays the current configuration register value, and asks whether you want to change it:

```
Current Configuration Register: 0x00000041
Configuration Summary:
  boot default image from Flash
  ignore system configuration

Do you wish to change this configuration? y/n [n]: y
```

Step 6 Record the current configuration register value, so you can restore it later.

Step 7 At the prompt, enter **Y** to change the value.

The ASA prompts you for new values.

Step 8 Accept the default values for all settings, except for the "disable system configuration?" value.

Step 9 At the prompt, enter **Y**.

Step 10 Reload the ASA by entering the following command:

```
rommon #2> boot
Launching BootLoader...
Boot configuration file contains 1 entry.

Loading disk0:/asa800-226-k8.bin... Booting...Loading...
```

The ASA loads the default configuration instead of the startup configuration.

Step 11 Access the privileged EXEC mode by entering the following command:

```
ciscoasa# enable
```

Step 12 When prompted for the password, press **Enter**.

The password is blank.

Step 13 Load the startup configuration by entering the following command:

```
ciscoasa# copy startup-config running-config
```

Step 14 Access the global configuration mode by entering the following command:

```
ciscoasa# configure terminal
```

Step 15 Change the passwords, as required, in the default configuration by entering the following commands:

```
ciscoasa(config)# password password
ciscoasa(config)# enable password password
ciscoasa(config)# username name password password
```

Step 16 Load the default configuration by entering the following command:

```
ciscoasa(config)# no config-register
```

The default configuration register value is 0x1. See the command reference for more information about the configuration register.

Step 17 Save the new passwords to the startup configuration by entering the following command:

```
ciscoasa(config)# copy running-config startup-config
```

Recover Passwords on the ASA 5506-X, ASA 5508-X, and ASA 5516-X

To recover passwords for the ASA 5506-X, ASA 5508-X, and ASA 5516-X perform the following steps:

Procedure

Step 1 Connect to the ASA console port.

Step 2 Power off the ASA, then power it on.

Step 3 After startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 To update the configuration register value, enter the following command:

```
rommon #1> confreg 0x41

You must reset or power cycle for new config to take effect
```

The ASA displays the current configuration register value and a list of configuration options. Record the current configuration register value, so you can restore it later.

```
Configuration Register: 0x00000041

Configuration Summary
 [ 0 ] password recovery
 [ 1 ] display break prompt
 [ 2 ] ignore system configuration
 [ 3 ] auto-boot image in disks
 [ 4 ] console baud: 9600
 boot: ...... auto-boot index 1 image in disks
```

Step 5 Reload the ASA by entering the following command:

```
rommon #2> boot
Launching BootLoader...
Boot configuration file contains 1 entry.

Loading disk0:/asa932-226-k8.bin... Booting...Loading...
```

The ASA loads the default configuration instead of the startup configuration.

Step 6 Access the privileged EXEC mode by entering the following command:

```
ciscoasa# enable
```

Step 7 When prompted for the password, press **Enter**.

The password is blank.

Step 8 Load the startup configuration by entering the following command:

```
ciscoasa# copy startup-config running-config
```

Step 9 Access the global configuration mode by entering the following command:

```
ciscoasa# configure terminal
```

Step 10 Change the passwords, as required, in the default configuration by entering the following commands:

```
ciscoasa(config)# password password
ciscoasa(config)# enable password password
ciscoasa(config)# username name password password
```

Step 11 Load the default configuration by entering the following command:

```
ciscoasa(config)# no config-register
```

The default configuration register value is 0x1. See the command reference for more information about the configuration register.

Step 12 Save the new passwords to the startup configuration by entering the following command:

```
ciscoasa(config)# copy running-config startup-config
```

Recover Passwords or Images on the ASAv

To recover passwords or images on the ASAv, perform the following steps:

Procedure

Step 1 Copy the running configuration to a backup file on the ASAv:

copy running-config *filename*

Example:

```
ciscoasa# copy running-config backup.cfg
```

Step 2 Restart the ASAv:

reload

Step 3 From the GNU GRUB menu, press the down arrow, choose the <**filename**> **with no configuration load** option, then press **Enter**. The filename is the default boot image filename on the ASAv. The default boot image is never automatically booted through the **fallback** command. Then load the selected boot image.

```
GNU GRUB version 2.0(12)4
bootflash:/asa100123-20-smp-k8.bin
bootflash: /asa100123-20-smp-k8.bin with no configuration load
```

Example:

```
GNU GRUB version 2.0(12)4
bootflash: /asa100123-20-smp-k8.bin with no configuration load
```

Step 4 Copy the backup configuration file to the running configuration.

copy *filename* **running-config**

Example:

```
ciscoasa (config)# copy backup.cfg running-config
```

Step 5 Reset the password.

enable password *password*

Example:

```
ciscoasa(config)# enable password cisco123
```

Step 6 Save the new configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

Disable Password Recovery for ASA Hardware

Note You cannot disable password recovery on the ASAv or Firepower models.

To disable password recovery to ensure that unauthorized users cannot use the password recovery mechanism to compromise the ASA, perform the following steps.

Before you begin

On the ASA, the **no service password-recovery** command prevents you from entering ROMMON mode with the configuration intact. When you enter ROMMON mode, the ASA prompts you to erase all Flash file systems. You cannot enter ROMMON mode without first performing this erasure. If you choose not to erase the Flash file system, the ASA reloads. Because password recovery depends on using ROMMON mode and maintaining the existing configuration, this erasure prevents you from recovering a password. However, disabling password recovery prevents unauthorized users from viewing the configuration or inserting different passwords. In this case, to restore the system to an operating state, load a new image and a backup configuration file, if available.

The **service password-recovery** command appears in the configuration file for information only. When you enter the command at the CLI prompt, the setting is saved in NVRAM. The only way to change the setting is to enter the command at the CLI prompt. Loading a new configuration with a different version of the command does not change the setting. If you disable password recovery when the ASA is configured to ignore the startup configuration at startup (in preparation for password recovery), then the ASA changes the setting to load the startup configuration as usual. If you use failover, and the standby unit is configured to ignore the startup configuration, then the same change is made to the configuration register when the **no service password-recovery** command replicates to the standby unit.

Procedure

Disable password recovery.

no service password-recovery

Example:

```
ciscoasa (config)# no service password-recovery
```

View Debugging Messages

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to use **debug** commands during periods of less network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use. To enable debugging messages, see the **debug** commands in the command reference.

Packet Capture

Capturing packets may be useful when troubleshooting connectivity problems or monitoring suspicious activity. We recommend that you contact Cisco TAC if you want to use the packet capture service.

Guidelines for Packet Capture

Context Mode

- You can configure captures on the cluster control link within a context; only the packet that is associated with the context sent in the cluster control link is captured.
- You can only configure one capture for a shared VLAN; if you configure a capture in multiple contexts on the shared VLAN, then only the last capture that was configured is used.
- If you remove the last-configured (active) capture, no captures become active, even if you have previously configured a capture in another context; you must remove the capture and add it again to make it active.
- All traffic that enters the interface to which the capture is attached is captured, including traffic to other contexts on the shared VLAN. Therefore, if you enable a capture in Context A for a VLAN that is also used by Context B, both Context A and Context B ingress traffic are captured.
- For egress traffic, only the traffic of the context with the active capture is captured. The only exception is when you do not enable the ICMP inspection (therefore the ICMP traffic does not have a session in the accelerated path). In this case, both ingress and egress ICMP traffic for all contexts on the shared VLAN is captured.

Additional Guidelines

- If the ASA receives packets with an incorrectly formatted TCP header and drops them because of the *invalid-tcp-hdr-length* ASP drop reason, the **show capture** command output on the interface where those packets are received does not show those packets.
- You can only capture IP traffic; you cannot capture non-IP packets such as ARPs.
- For inline SGT tagged packets, captured packets contain an additional CMD header that your PCAP viewer might not understand.
- Packet captures include packets that the system modifies or injects into the connection due to inspection, NAT, TCP normalization, or other features that adjust the content of a packet.
- The trace of the lifespan of an injected virtual packet in a datapath does not exactly reflect how the datapath handles the physical packets. This difference depends on the software version, configuration, and type of the injected virtual packets. Following are configuration settings that might lead to the disparity:
  - at least 2 NAT statements for the same host exist.
  - forward and reverse flows of a connection having different protocols. For example, forward flow is UDP or TCP, reverse flow is ICMP.
  - ICMP error inspection being enabled.

Capture Packets

To capture packets, perform the following steps.

Procedure

Step 1 Enable packet capture capabilities for packet sniffing and network fault isolation:

capture *capture_name* [**type** {**asp-drop** [**all** | *drop-code*] | **tls-proxy** | **raw-data** | **isakmp** [**ikev1** | **ikev2**] | **inline-tag** [*tag*] | **webvpn user** *webvpn-user*}] [**access-list** *access_list_name*] {**interface** {*interface_name* | **asa_dataplane** | **asa_mgmt_plane** | **cplane**} } [**buffer** *buf_size*] [**ethernet-type** *type*] [**reeinject-hide**] [**packet-length** *bytes*] [**circular-buffer**] [**trace** [**trace-count** *number*]] [**real-time** [**dump**] [**detail**]] [**file-size**] [**headers-only**] [**match** *protocol* {**host** *source-ip* | *source-ip mask* | **any**} [*operator src_port*] {**host** *dest_ip* |*dest_ip mask* | **any**} [*operator dest_port*]]

Example:

```
ciscoasa# capture captest interface inside
```

You must configure an interface for any packets to be captured. Use the same *capture_name* on multiple **capture** statements to capture multiple types of traffic.

The **type asp-drop** keyword captures packets dropped by the accelerated security path. In a cluster, dropped forwarded data packets from one unit to another are also captured. In multiple context mode, when this option is issued in the system execution space, all dropped data packets are captured; when this option is issued in a context, only dropped data packets that enter from interfaces belonging to the context are captured.

The **type raw-data** keywords capture inbound and outbound packets. This setting is the default.

The **inline-tag** *tag* keyword-argument pair specifies a tag for a particular SGT value or leaves it unspecified to capture a tagged packet with any SGT value.

The **buffer** keyword defines the buffer size used to store the packet. When the byte buffer is full, packet capture stops. When used in a cluster, this is the per-unit size, not the sum of all units. The **circular-buffer** keyword overwrites the buffer, starting from the beginning, when the buffer is full.

The **ethernet-type** keyword sets an ethernet type to capture. Supported Ethernet types include 8021Q, ARP, IP, IP6, LACP, PPPOED, PPPOES, RARP, and VLAN. An exception occurs with the 802.1Q or VLAN type. The 802.1Q tag is automatically skipped and the inner Ethernet type is used for matching. IP is the default ethernet type.

The **interface** keyword sets the name of the interface on which to use packet capture.

To capture packets on the dataplane, use the **asa_dataplane** keyword. To filter packets captured on an add-on module backplane, use the **asa_dataplane** option and follow these guidelines: In single mode, the backplane control packets bypass the access list and are captured. In multiple context mode, only control packets are captured in the system execution space. Data packets are captured in the context.

To configure the size of capture file, use the **file-size**keyword. The file size can be between 32 and 10000 MB.

If you want to capture only L2, L3 and L4 headers of packet without data in them, use the **headers-only** command.

The **match** keyword captures matching the protocol and source and destination IP addresses and optional ports. You can use this keyword up to three times in one command. The **any** keyword captures IPv4 traffic only. The operator can be one of the following:

- lt—less than
- gt—greater than
- eq—equal to

The **real-time** keyword displays the captured packets continuously in real-time.

The **reinject-hide** keyword specifies that no reinjected packets will be captured and applies only in a clustering environment.

Note If ACL optimization is configured, you cannot use the **access-list** command in capture. You can only use the **access-group** command. An error appears if you try to use the **access-list** command in this case.

Step 2 Capture cluster control-link traffic:

capture *capture_name* {**type lacp interface** *interface_id* [**buffer** *buf_size*] [**packet-length** *bytes*] [**circular-buffer**] [**real-time** [**dump**] [**detail**]]

capture *capture_name* **interface cluster** [**buffer** *buf_size*] [**ethernet-type** *type*] [**packet-length** *bytes*] [**circular-buffer**] [**trace** [**trace-count** *number*]] [**real-time** [**dump**] [**detail**]] [**trace**] [**match** *protocol* {**host** *source-ip* | *source-ip mask* | **any**} [*operator src_port*] {**host** *dest_ip* |*dest_ip mask* | **any**} [*operator dest_port*]]

Example:

```
ciscoasa# capture ccl type lacp interface GigabitEthernet0/0
ciscoasa# capture ccl interface cluster match udp any eq 49495 any
ciscoasa# capture ccl interface cluster match udp any any eq 49495
```

You can capture cluster control link traffic in two ways: to capture all the traffic on the cluster control link, use the **cluster** keyword for the interface name. To capture only cLACP packets, specify **type lacp**, and specify the physical interface ID instead of the interface name. There are two types of packets on the cluster control link: control plane packets and data plane packets, which both include forwarded data traffic and cluster LU messages. The TTL field in the IP address header is encoded to differentiate between these two types of packets. When forwarded data packets are captured, their clustering trailers are included in the capture file for debugging purposes.

Step 3 Capture packets cluster-wide:

cluster exec capture *capture_name arguments*

Step 4 Stop the packet capture:

no capture *capture_name*

To terminate a real-time packet capture, enter **Ctrl + c.** To permanently remove the capture, use the **no** form of this command. The real-time option applies only to **raw-data** and **asp-drop** captures.

Step 5 Clear the capture:

clear capture *capture_name*

Examples

Control Plane Packets

All packets to and from the control plane have a TTL of 255, and port number 49495 is used for the clustering control-plane listen port. The following example shows how to create a LACP capture for the clustering environment:

```
ciscoasa# capture lacp type lacp interface GigabitEthernet0/0
```

The following example shows how to create a capture for control path packets in the clustering link:

```
ciscoasa# capture cp interface cluster match udp any eq 49495 any
ciscoasa# capture cp interface cluster match udp any any eq 49495
```

Data Plane Packets

Data packets include those forwarded from one unit to another unit (its connection owner) and cluster LU messages. Regular cluster LU update messages have a TTL of 254, and there is a special LU packet that has a TTL of 253. This special LU packet is only for TCP, and it only happens when the director elects a new flow owner; the director sends back the requesting packet along with the CLU_FULL update packet. The LU packet is filled with the original packet's L3/L4 header to avoid a potential race condition at the receiver side. Forwarded data packets have a TTL of less than 4. The following example shows how to create a capture for data path packets in the cluster control link. To capture all inter-cluster dataplane "flow logical update" messages, use port 4193.

```
ciscoasa# access-list cc1 extended permit udp any any eq 4193
ciscoasa# access-list cc1 extended permit udp any eq 4193 any
ciscoasa# capture dp interface cluster access-list ccl
```

View a Packet Capture

You can view a packet capture at the CLI, in a browser, or download a capture to a server of your choice.

Procedure

Step 1 View the capture at the CLI:

[**cluster exec**] **show capture** [*capture_name*] [**access-list** *access_list_name*] [**count** *number*] [**decode**] [**detail**] [**dump**] [**packet-number** *number*]

Example:

```
ciscoasa# show capture capin

 8 packets captured

 1: 03:24:35.526812       192.168.10.10 > 203.0.113.3: icmp: echo request
 2: 03:24:35.527224       203.0.113.3 > 192.168.10.10: icmp: echo reply
 3: 03:24:35.528247       192.168.10.10 > 203.0.113.3: icmp: echo request
 4: 03:24:35.528582       203.0.113.3 > 192.168.10.10: icmp: echo reply
 5: 03:24:35.529345       192.168.10.10 > 203.0.113.3: icmp: echo request
 6: 03:24:35.529681       203.0.113.3 > 192.168.10.10: icmp: echo reply
 7: 03:24:57.440162       192.168.10.10 > 203.0.113.3: icmp: echo request
 8: 03:24:57.440757       203.0.113.3 > 192.168.10.10: icmp: echo reply
```

The **access-list** keyword displays information for packets that are based on IP or higher fields for the specific access list identification.

The **cluster exec** keyword lets you issue the **show capture** command in one unit and run the command in all the other units at the same time.

The **count** keyword displays the number of packets specified data.

The **decode** keyword is useful when a capture of type **isakmp** is applied to an interface. All ISAKMP data flowing through that interface will be captured after decryption and shown with more information after decoding the fields.

The **detail** keyword displays additional protocol information for each packet.

The **dump** keyword displays a hexadecimal dump of the packets that are transported over the data link.

The **packet-number** keyword starts the display at the specified packet number.

Step 2 View the packet capture with your browser:

https://*ip_of_asa***/admin/capture/***capture_name***/pcap**

If you leave out the **pcap** keyword, then only the equivalent of the **show capture** *capture_name* command output is provided.

In multiple context mode, the **copy capture** command is available only in the system execution space.

Step 3 Copy the packet capture to a server. This example shows FTP.

[**cluster exec**] **copy /pcap capture:**[*context-name/*]*capture_name* **ftp://***username***:***password***@***server_ip***/***path*

If you leave out the **pcap** keyword, then only the equivalent of the **show capture** *capture_name* command output is provided.

Note When you copy a packet capture to a disk, ensure that the capture filename is less than or equal to 63 characters. When the filename is more than 63 characters, though the packet capture is successful, copying the capture to a disk fails.

Examples

The following example shows a type asp-drop capture:

```
ciscoasa# capture asp-drop type asp-drop acl-drop
ciscoasa# show capture asp-drop

 2 packets captured

 1: 04:12:10.428093       192.168.10.10.34327 > 10.94.0.51.15868: S
    2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
    Flow is denied by configured rule
 2: 04:12:12.427330       192.168.10.10.34327 > 10.94.0.51.15868: S
```

```
    2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
    Flow is denied by configured rule
 2 packets shown

ciscoasa# show capture asp-drop

 2 packets captured

 1: 04:12:10.428093       192.168.10.10.34327 > 10.94.0.51.15868: S
    2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
    Flow is denied by configured rule
 2: 04:12:12.427330       192.168.10.10.34327 > 10.94.0.51.15868: S
    2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
    Flow is denied by configured rule
 2 packets shown
```

The following example shows an ethernet-type capture:

```
ciscoasa# capture arp ethernet-type arp interface inside
ciscoasa# show cap arp

22 packets captured

   1: 05:32:52.119485       arp who-has 10.10.3.13 tell 10.10.3.12
   2: 05:32:52.481862       arp who-has 192.168.10.123 tell 192.168.100.100
   3: 05:32:52.481878       arp who-has 192.168.10.50 tell 192.168.100.10
   4: 05:32:53.409723       arp who-has 10.106.44.135 tell 10.106.44.244
   5: 05:32:53.772085       arp who-has 10.106.44.108 tell 10.106.44.248
   6: 05:32:54.782429       arp who-has 10.106.44.135 tell 10.106.44.244
   7: 05:32:54.784695       arp who-has 10.106.44.1 tell 11.11.11.112:
```

View the Crash Dump

If the ASA or ASAv crashes, you can view the crash dump information. We recommend that you contact Cisco TAC if you want to interpret the crash dump. See the **show crashdump** command in the command reference.

View the Coredump

A coredump is a snapshot of the running program when the program has terminated abnormally or crashed. Coredumps are used to diagnose or debug errors and save a crash for future off-site analysis. Cisco TAC may request that you enable the coredump feature to troubleshoot application or system crashes on the ASA or ASAv. See the **coredump** command in the command reference.

vCPU Usage in the ASAv

The ASAv vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The vSphere reported vCPU usage includes the ASAv usage as described plus:

- ASAv idle time
- %SYS overhead used for the ASAv VM
- Overhead of moving packets between vSwitches, vNICs, and pNICs. This overhead can be quite significant.

CPU Usage Example

The following is an example in which the reported vCPU usage is substantially different:

- ASAv reports: 40%
- DP: 35%
- External Processes: 5%
- vSphere reports: 95%
- ASA (as ASAv reports): 40%
- ASA idle polling: 10%
- Overhead: 45%

The overhead is used to perform hypervisor functions and to move packets between NICs and vNICs using the vSwitch.

Usage can exceed 100% because the ESXi server can use additional compute resources for overhead on behalf of the ASAv.

VMware CPU Usage Reporting

In vSphere, click the **VM Performance** tab, then click **Advanced** to display the **Chart Options** drop-down list, which shows vCPU usage for each state (%USER, %IDLE, %SYS, and so on) of the VM. This information is useful for understanding VMware's perspective on where CPU resources are being used.

On the ESXi server shell (you access the shell by using SSH to connect to the host), esxtop is available. Esxtop has a similar look and feel to the Linux **top** command and provides VM state information for vSphere performance, including the following:

- Details on vCPU, memory, and network usage
- vCPU usage for each state of each VM.
- Memory (type M while running) and network (type N while running), as well as statistics and the number of RX drops

ASAv and vCenter Graphs

There are differences in the CPU % numbers between the ASAv and vCenter:

- The vCenter graph numbers are always higher than the ASAv numbers.
- vCenter calls it %CPU usage; the ASAv calls it %CPU utilization.

The terms “%CPU utilization” and “%CPU usage” mean different things:

- CPU utilization provides statistics for physical CPUs.
- CPU usage provides statistics for logical CPUs, which is based on CPU hyperthreading. But because only one vCPU is used, hyperthreading is not turned on.

vCenter calculates the CPU % usage as follows:

Amount of actively used virtual CPUs, specified as a percentage of the total available CPUs

This calculation is the host view of the CPU usage, not the guest operating system view, and is the average CPU utilization over all available virtual CPUs in the virtual machine.

For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. The virtual CPU usage calculation is Usage in MHz / number of virtual CPUs x core frequency

When you compare the usage in MHz, both the vCenter and ASAv numbers match. According to the vCenter graph, MHz % CPU usage is calculated as 60/(2499 x 1 vCPU) = 2.4

Test Your Configuration

This section describes how to test connectivity for the single mode ASA or for each security context, how to ping the ASA interfaces, and how to allow hosts on one interface to ping through to hosts on another interface.

Test Basic Connectivity: Pinging Addresses

Ping is a simple command that lets you determine if a particular address is alive and responsive. The following topics explain more about the command and what types of testing you can accomplish with it.

What You Can Test Using Ping

When you ping a device, a packet is sent to the device and the device returns a reply. This process enables network devices to discover, identify, and test each other.

You can use ping to do the following tests:

- Loopback testing of two interfaces—You can initiate a ping from one interface to another on the same ASA, as an external loopback test to verify basic “up” status and operation of each interface.
- Pinging to an ASA—You can ping an interface on another ASA to verify that it is up and responding.
- Pinging through an ASA—You can ping through an intermediate ASA by pinging a device on the other side of the ASA. The packets will pass through two of the intermediate ASA’s interfaces as they go in each direction. This action performs a basic test of the interfaces, operation, and response time of the intermediate unit.
- Pinging to test questionable operation of a network device—You can ping from an ASA interface to a network device that you suspect is functioning incorrectly. If the interface is configured correctly and an echo is not received, there might be problems with the device.
- Pinging to test intermediate communications—You can ping from an ASA interface to a network device that is known to be functioning correctly. If the echo is received, the correct operation of any intermediate devices and physical connectivity is confirmed.

Choosing Between ICMP and TCP Ping

The ASA includes the traditional ping, which sends ICMP Echo Request packets and gets Echo Reply packets in return. This is the standard tool and works well if all intervening network devices allow ICMP traffic. With ICMP ping, you can ping IPv4 or IPv6 addresses, or host names.

However, some networks prohibit ICMP. If this is true of your network, you can instead use TCP ping to test network connectivity. With TCP ping, the ping sends TCP SYN packets, and considers the ping a success if it receives a SYN-ACK in response. With TCP ping, you can ping IPv4 addresses or host names, but you cannot ping IPv6 addresses.

Keep in mind that a successful ICMP or TCP ping simply means that the address you are using is alive and responding to that specific type of traffic. This means that basic connectivity is working. Other policies running on a device could prevent specific types of traffic from successfully getting through a device.

Enable ICMP

By default, you can ping from a high security interface to a low security interface. You just need to enable ICMP inspection to allow returning traffic through. If you want to ping from low to high, then you need to apply an ACL to allow traffic.

When pinging an ASA interface, any ICMP rules applied to the interface must allow Echo Request and Echo Response packets. ICMP rules are optional: if you do not configure them, all ICMP traffic to an interface is allowed.

This procedure explains all of ICMP configuration you might need to complete to enable ICMP pinging of ASA interfaces, or for pinging through an ASA.

Procedure

Step 1 Ensure ICMP rules allow Echo Request/Echo Response.

ICMP rules are optional and apply to ICMP packets sent directly to an interface. If you do not apply ICMP rules, all ICMP access is allowed. In this case, no action is required.

However, if you do implement ICMP rules, ensure that you include at least the following on each interface, replacing “inside” with the name of an interface on your device.

```
ciscoasa(config)# icmp permit 0.0.0.0 0.0.0.0 echo inside
ciscoasa(config)# icmp permit 0.0.0.0 0.0.0.0 echo-reply inside
```

Step 2 Ensure access rules allow ICMP.

When pinging a host through an ASA, access rules must allow ICMP traffic to leave and return. The access rule must at least allow Echo Request/Echo Reply ICMP packets. You can add these rules as global rules.

Assuming you already have access rules applied to interfaces or applied globally, simply add these rules to the relevant ACL, for example:

```
ciscoasa(config)# access-list outside_access_in extendedpermit icmp any anyecho

ciscoasa(config)# access-list outside_access_in extendedpermit icmp any
anyecho-reply
```

Alternatively, just allow all ICMP:

```
ciscoasa(config)# access-list outside_access_in extendedpermit icmp any any
```

If you do not have access rules, you will need to also allow the other type of traffic you want, because applying any access rules to an interface adds an implicit deny, so all other traffic will be dropped. Use the **access-group** command to apply the ACL to an interface or globally.

If you are simply adding the rule for testing purposes, you can use the **no** form of the **access-list** command to remove the rule from the ACL. If the entire ACL is simply for testing purposes, use the **no access-group** command to remove the ACL from the interface.

Step 3 Enable ICMP inspection.

ICMP inspection is needed when pinging through the ASA, as opposed to pinging an interface. Inspection allows returning traffic (that is, the Echo Reply packet) to return to the host that initiated the ping, and also ensures there is one response per packet, which prevents certain types of attack.

You can simply enable ICMP inspection in the default global inspection policy.

```
ciscoasa(config)# policy-map  global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect icmp
```

Ping Hosts

To ping any device, you simply enter **ping** with the IP address or host name, such as **ping 10.1.1.1** or **ping www.example.com**. For TCP ping, you include the **tcp** keyword and the destination port, such as **ping tcp www.example.com 80**. That is usually the extent of any test you need to run.

Example output for a successful ping:

```
Sending 5, 100-byte ICMP Echos to out-pc, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

If the ping fails, the output indicates ? for each failed attempt, and the success rate is less than 100 percent (complete failure is 0 percent):

```
Sending 5, 100-byte ICMP Echos to 10.132.80.101, timeout is 2 seconds:
?????
Success rate is 0 percent (0/5)
```

However, you can also add parameters to control some aspects of the ping. Following are your basic options:

- ICMP ping.

  ping [*if_name*] *host* [**repeat** *count*] [**timeout** *seconds*] [**data** *pattern*] [**size** *bytes*] [**validate**]

  Where:

  - *if_name* is the name of the interface by which the host is accessible. If you do not include a name, the routing table is used to determine the interface to use.
  - *host* is the IPv4, IPv6, or host name of the host you are pinging.
  - **repeat** *count* is how many packets to send. The default is 5.
  - **timeout** *seconds* is the number of seconds for each packet to time out if no response occurs. The default is 2.
  - **data** *pattern* is the hexadecimal pattern to use in the packets sent. The default is 0xabcd.
  - **size** *bytes* is the length of the packet sent. The default is 100 bytes.
  - **validate** indicates that you want reply data validated.

- TCP ping.

  ping tcp [*if_name*] *host* [*port*] [**repeat** *count*] [**timeout** *seconds*] [**source** *host* [*ports*]

  Where:

  - *if_name* is the interface through which the source sends the ping. If you do not include a name, the routing table is used.
  - *host* is the IPv4 address or host name of the destination you are pinging. You cannot use TCP ping with IPv6 addresses.
  - *port* is the TCP port on the host you are pinging.
  - **repeat** and **timeout** have the same meaning as above.
  - **source** *host port* indicates the source host and port for the ping. Use port 0 to get a random port.

- Interactive ping.

  ping

  By entering ping without parameters, you are prompted for interface, destination, and other parameters, including extended parameters not available as keywords. Use this method if you have need for extensive control over the ping packets.

Test ASA Connectivity Systematically

If you want to do a more systematic test of ASA connectivity, you can use the following general procedure.

Before you begin

If you want to see the syslog messages mentioned in the procedure, enable logging (the **logging enable** command, or **Configuration > Device Management > Logging > Logging Setup** in ASDM).

Although unnecessary, you can also enable ICMP debug to see messages on the ASA console as you ping ASA interfaces from external devices (you will not see debug messages for pings that go through the ASA). We recommend that you only enable pinging and debugging messages during troubleshooting, as they can affect performance. The following example enables ICMP debugging, sets syslog messages to be sent to Telnet or SSH sessions and sends them to those sessions, and enables logging. Instead of using the **logging monitor debug** command, you can alternately use the **logging buffer debug** command to send log messages to a buffer, and then view them later using the **show logging**command.

```
ciscoasa(config)# debug icmp trace
ciscoasa(config)# logging monitor debug
ciscoasa(config)# terminal monitor
ciscoasa(config)# logging enable
```

With this configuration, you would see something like the following for a successful ping from an external host (209.165.201.2) to the ASA outside interface (209.165.201.1):

```
ciscoasa(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

The output shows the ICMP packet length (32 bytes), the ICMP packet identifier (1), and the ICMP sequence number (the ICMP sequence number starts at 0, and is incremented each time that a request is sent).

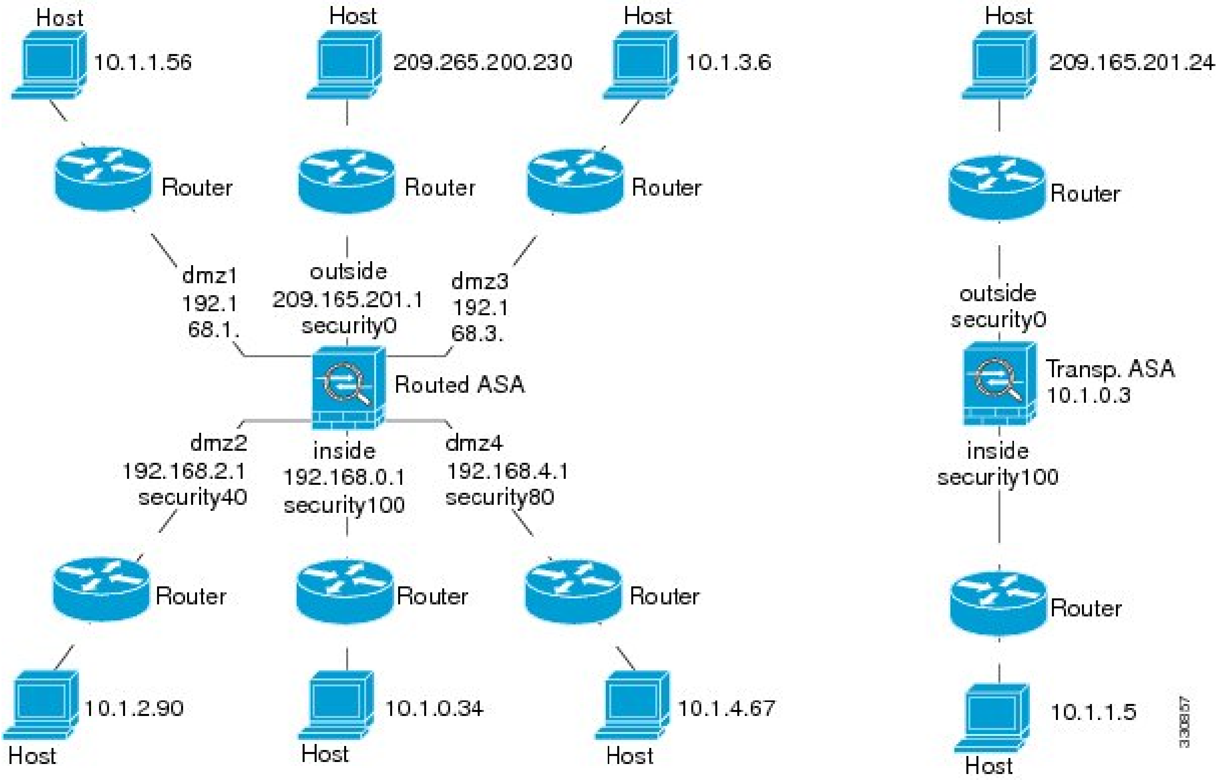
When you are finished testing, disable debugging. Leaving the configuration in place can pose performance and security risks. If you enabled logging just for testing, you can disable it also.

```
ciscoasa(config)# no debug icmp trace
ciscoasa(config)# no logging monitor debug
ciscoasa(config)# no terminal monitor
ciscoasa(config)# no logging enable
```

Procedure

Step 1 Draw a diagram of your single-mode ASA or security context that shows the interface names, security levels, and IP addresses. The diagram should also include any directly connected routers and a host on the other side of the router from which you will ping the ASA.

Figure 59: Network Diagram with Interfaces, Routers, and Hosts



Step 2 Ping each ASA interface from the directly connected routers. For transparent mode, ping the BVI IP address. This test ensures that the ASA interfaces are active and that the interface configuration is correct.

A ping might fail if the ASA interface is not active, the interface configuration is incorrect, or if a switch between the ASA and a router is down (see the following figure). In this case, no debugging messages or syslog messages appear, because the packet never reaches the ASA.

Figure 60: Ping Failure at the ASA Interface

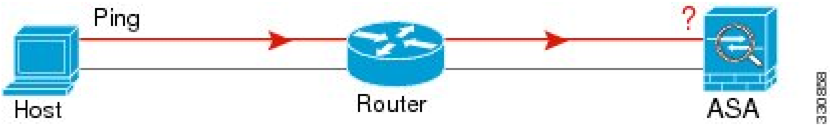
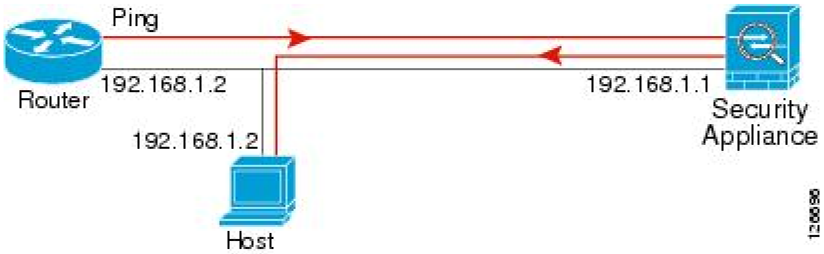


Figure 61: Ping Failure Because of IP Addressing Problems

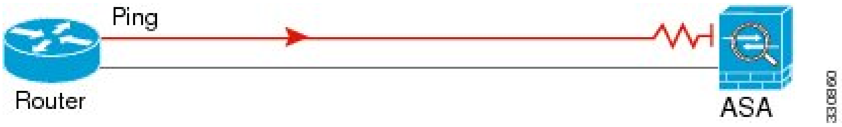


If the ping reply does not return to the router, then a switch loop or redundant IP addresses might exist (see the following figure).

Step 3 Ping each ASA interface from a remote host. For transparent mode, ping the BVI IP address. This test checks whether the directly connected router can route the packet between the host and the ASA, and whether the ASA can correctly route the packet back to the host.

A ping might fail if the ASA does not have a return route to the host through the intermediate router (see the following figure). In this case, the debugging messages show that the ping was successful, but syslog message 110001 appears, indicating a routing failure has occurred.

Figure 62: Ping Failure Because the ASA Has No Return Route



Step 4 Ping from an ASA interface to a network device that you know is functioning correctly.

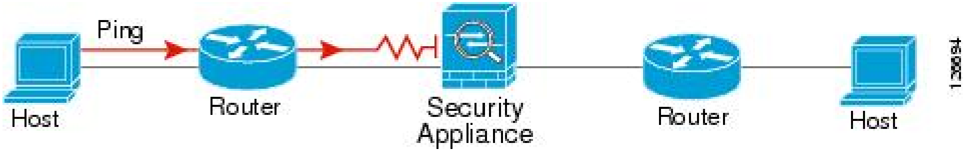
- If the ping is not received, a problem with the transmitting hardware or interface configuration may exist.
- If the ASA interface is configured correctly and it does not receive an echo reply from the “known good” device, problems with the interface hardware receiving function may exist. If a different interface with “known good” receiving capability can receive an echo after pinging the same “known good” device, the hardware receiving problem of the first interface is confirmed.

Step 5 Ping from the host or router through the source interface to another host or router on another interface. Repeat this step for as many interface pairs as you want to check. If you use NAT, this test shows that NAT is operating correctly.

If the ping succeeds, a syslog message appears to confirm the address translation for routed mode (305009 or 305011) and that an ICMP connection was established (302020). You can also enter either the **show xlate** or **show conns** command to view this information.

The ping might fail because NAT is not configured correctly. In this case, a syslog message appears, showing that the NAT failed (305005 or 305006). If the ping is from an outside host to an inside host, and you do not have a static translation, you get message 106010.

Figure 63: Ping Failure Because the ASA is Not Translating Addresses



Trace Routes to Hosts

If you are having problems sending traffic to an IP address, you can trace the route to the host to determine if there is a problem on the network path.

Procedure

Step 1 Make the ASA Visible on Trace Routes, on page 949.

Step 2 Determine Packet Routes, on page 950.

Make the ASA Visible on Trace Routes

By default, the ASA does not appear on traceroutes as a hop. To make it appear, you need to decrement the time-to-live on packets that pass through the ASA, and increase the rate limit on ICMP unreachable messages.

Procedure

Step 1 Create an L3/L4 class map to identify the traffic for which you want to customize connection settings.

class-map *name*

match *parameter*

Example:

```
ciscoasa(config)# class-map CONNS
ciscoasa(config-cmap)# match any
```

For information on matching statements, see the Service Policy chapter in the firewall configuration guide.

Step 2 Add or edit a policy map that sets the actions to take with the class map traffic, and identify the class map.

policy-map *name* **class** *name*

Example:

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class CONNS
```

In the default configuration, the global_policy policy map is assigned globally to all interfaces. If you want to edit the global_policy, enter global_policy as the policy name. For the class map, specify the class you created earlier in this procedure.

Step 3 Decrement time-to-live (TTL) on packets that match the class.

```
set connection decrement-ttl
```

Step 4 If you are editing an existing service policy (such as the default global policy called global_policy), you can skip this step. Otherwise, activate the policy map on one or more interfaces.

```
service-policy policymap_name {global | interface interface_name      }
```

Example:

```
ciscoasa(config)# service-policy global_policy global
```

The **global** keyword applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Step 5 Increase the rate limit on ICMP Unreachable messages so that the ASA will appear on trace route output.

icmp unreachable rate-limit *rate* **burst-size** *size*

Example:

```
ciscoasa(config)# icmp unreachable rate-limit 50 burst-size 1
```

The rate limit can be 1-100, with 1 being the default. The burst size is meaningless, but must be 1-10.

Example

The following example decrements TTL for all traffic globally and increase the ICMP unreachable limit to 50.

```
ciscoasa(config)# class-map global-policy
ciscoasa(config-cmap)# match any
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class global-policy
ciscoasa(config-pmap-c)# set connection decrement-ttl
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# icmp unreachable rate-limit 50 burst-size 6
```

Determine Packet Routes

Use Traceroute to help you to determine the route that packets will take to their destination. A traceroute works by sending UDP packets to a destination on an invalid port. Because the port is not valid, the routers along the way to the destination respond with an ICMP Time Exceeded Message, and report that error to the ASA.

The traceroute shows the result of each probe sent. Every line of output corresponds to a TTL value in increasing order. The following table explains the output symbols.

Output Symbol	Description
*	No response was received for the probe within the timeout period.
nn msec	For each node, the round-trip time (in milliseconds) for the specified number of probes.
!N.	ICMP network unreachable.

Output Symbol	Description
!H	ICMP host unreachable.
!P	ICMP unreachable.
!A	ICMP administratively prohibited.
?	Unknown ICMP error.

Before you begin

Traceroute does not support IPv6.

Procedure

Trace the route to a destination:

traceroute [*destination_ip* | *hostname*} [**source** {*source_ip* | *source-interface*}] [**numeric**] [**timeout** *timeout_value*] [**probe** *probe_num*] [**ttl** *min_ttl max_ttl*] [**port** *port_value*] [**use-icmp**]

Example:

```
ciscoasa# traceroute 209.165.200.225

Type escape sequence to abort.
Tracing the route to 209.165.200.225

1 10.83.194.1 0 msec 10 msec 0 msec
2 10.83.193.65 0 msec 0 msec 0 msec
3 10.88.193.101 0 msec 10 msec 0 msec
4 10.88.193.97 0 msec 0 msec 10 msec
5 10.88.239.9 0 msec 10 msec 0 msec
6 10.88.238.65 10 msec 10 msec 0 msec
7 172.16.7.221 70 msec 70 msec 80 msec
8 209.165.200.225 70 msec 70 msec 70 msec
```

Normally, you simply include the destination IP address or hostname, such as **traceroute www.example.com**. However, you can adjust the characteristics of the trace if desired:

- **source** {*source_ip* | *source-interface*}—Specifies the interface to use as the source of the trace. You can specify the interface by name or by IP address. In transparent mode, you must use the management address.
- **numeric**—Indicates that only the IP addresses should be shown in the trace route. Without this keyword, the trace route does DNS lookups for addresses and includes DNS names, assuming that you configure DNS.
- **timeout** *timeout_value*—How long to wait for a response before timing out. The default is 3 seconds.
- **probe** *probe_num*—How many probes to send at each TTL level. The default is 3.
- **ttl** *min_ttl max_ttl*—The minimum and maximum time-to-live values for the probes. The minimum default is one, but you can set it to a higher value to suppress the display of known hops. The maximum

default is 30. The traceroute terminates when the packet reaches the destination or when the maximum value is reached.

- **port** *port_value*—The UDP port to use. The default is 33434.
- **use-icmp**—Send ICMP packets instead of UDP packets for probes.

Using the Packet Tracer to Test Policy Configuration

You can test your policy configuration by modeling a packet based on source and destination addressing and protocol characteristics. The trace does policy lookup to test access rules, NAT, routing, and so forth, to see if the packet would be permitted or denied.

By testing packets this way, you can see the results of your policies and test whether the types of traffic you want to allow or deny are handled as desired. Besides verifying your configuration, you can use the tracer to debug unexpected behavior, such as packets being denied when they should be allowed.

Procedure

Step 1 The command is complicated, so we divided it into parts. Start by choosing the interface and protocol for the trace:

packet-tracer input *ifc_name* {**icmp** | **tcp** | **udp** | **rawip**} [**inline-tag** *tag*] ...

Where:

- **input** *ifc_name*—The name of the interface from which to start the trace.
- **icmp**, **tcp**, **udp**, **rawip**—The protocol to use. “rawip” is raw IP, that is, IP packets that are not TCP/UDP.
- **inline-tag** *tag*—(Optional.) The security group tag value embedded in the Layer 2 CMD header. Valid values range from 0 - 65533.

Step 2 Next, type in the source address and protocol criteria.

...{*src_ip* | **user** *username* | **security-group** {**name** *name* | **tag** *tag*} | **fqdn** *fqdn-string*}...

Where:

- *src_ip*—The source IPv4 or IPv6 address for the packet trace.
- **user** *username*—The user identity in the format of domain\user. The most recently mapped address for the user (if any) is used in the trace.
- **security-group** {**name** *name* | **tag** *tag*}—The source security group based on the IP-SGT lookup for Trustsec. You can specify a security group name or a tag number.
- **fqdn** *fqdn-string*—The fully qualified domain name of the source host, IPv4 only.

Step 3 Next, type in the protocol characteristics.

- ICMP—Enter the ICMP type (1-255), ICMP code (0-255), and optionally, the ICMP identifier. You must use numbers for each variable, for example, 8 for echo.

type code... [*ident*]...

- TCP/UDP—Enter the source port number.

  ...*src_port* ...

- Raw IP—Enter the protocol number, 0-255.

  ... *protocol* ...

Step 4 Finally, type in the destination address criteria, destination port for TCP/UDP traces, and optional keywords, and press **Enter**.

... {*dst_ip* | **security-group** {**name** *name* | **tag** *tag*} | **fqdn** *fqdn-string*} *dst_port* [**detailed**] [**xml**]

Where:

- *dst_ip*—The destination IPv4 or IPv6 address for the packet trace.
- **security-group** {**name** *name* **|** **tag** *tag*}—The destination security group based on the IP-SGT lookup for Trustsec. You can specify a security group name or a tag number.
- **fqdn** *fqdn-string*—The fully qualified domain name of the destination host, IPv4 only.
- *dst_port*—The destination port for TCP/UDP traces. Do not include this value for ICMP or raw IP traces.
- **detailed**—Provides detailed trace results information in addition to the normal output.
- **xml**—Displays the trace results in XML format.

Example

The following example traces a TCP packet for the HTTP port from 10.100.10.10 to 10.100.11.11. The result indicates that the packet will be dropped by the implicit deny access rule.

```
ciscoasa(config)# packet-tracer input outside tcp 10.100.10.10 80 10.100.11.11
 80

Phase: 1
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 10.86.116.1 using egress ifc  outside

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: DROP
Config:
Implicit Rule
Additional Information:

Result:
input-interface: outside
input-status: up
input-line-status: up
```

```
output-interface: NP Identity Ifc
output-status: up
output-line-status: up
Action: drop
Drop-reason: (acl-drop) Flow is denied by configured rule
```

Monitoring Connections

To view current connections with information about source, destination, protocol, and so forth, use the **show conn all detail** command.

PART VIII

Monitoring

CHAPTER 40

Logging

This chapter describes how to log system messages and use them for troubleshooting.

About Logging

System logging is a method of collecting messages from devices to a server running a syslog daemon. Logging to a central syslog server helps in aggregation of logs and alerts. Cisco devices can send their log messages to a UNIX-style syslog service. A syslog service accepts messages and stores them in files, or prints them according to a simple configuration file. This form of logging provides protected long-term storage for logs. Logs are useful both in routine troubleshooting and in incident handling.

The ASA system logs provide you with information for monitoring and troubleshooting the ASA. With the logging feature, you can do the following:

- Specify which syslog messages should be logged.
- Disable or change the severity level of a syslog message.
- Specify one or more locations where syslog messages should be sent, including:
  - An internal buffer
  - One or more syslog servers
  - ASDM
  - An SNMP management station
  - Specified e-mail addresses
  - Console
  - Telnet and SSH sessions.

- Configure and manage syslog messages in groups, such as by severity level or class of message.
- Specify whether or not a rate-limit is applied to syslog generation.
- Specify what happens to the contents of the internal log buffer when it becomes full: overwrite the buffer, send the buffer contents to an FTP server, or save the contents to internal flash memory.
- Filter syslog messages by locations, severity level, class, or a custom message list.

Logging in Multiple Context Mode

Each security context includes its own logging configuration and generates its own messages. If you log in to the system or admin context, and then change to another context, messages you view in your session are only those messages that are related to the current context.

Syslog messages that are generated in the system execution space, including failover messages, are viewed in the admin context along with messages generated in the admin context. You cannot configure logging or view any logging information in the system execution space.

You can configure the ASA to include the context name with each message, which helps you differentiate context messages that are sent to a single syslog server. This feature also helps you to determine which messages are from the admin context and which are from the system; messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID.

Syslog Message Analysis

The following are some examples of the type of information you can obtain from a review of various syslog messages:

- Connections that are allowed by ASA security policies. These messages help you spot holes that remain open in your security policies.
- Connections that are denied by ASA security policies. These messages show what types of activity are being directed toward your secured inside network.
- Using the ACE deny rate logging feature shows attacks that are occurring on your ASA.
- IDS activity messages can show attacks that have occurred.
- User authentication and command usage provide an audit trail of security policy changes.
- Bandwidth usage messages show each connection that was built and torn down as well as the duration and traffic volume used.
- Protocol usage messages show the protocols and port numbers used for each connection.
- Address translation audit trail messages record NAT or PAT connections being built or torn down, which are useful if you receive a report of malicious activity coming from inside your network to the outside world.

Syslog Message Format

Syslog messages begin with a percent sign (%) and are structured as follows:

```
%ASA Level Message_number: Message_text
```

Field descriptions are as follows:

ASA	The syslog message facility code for messages that are generated by the ASA. This value is always ASA.
Level	1 through 7. The level reflects the severity of the condition described by the syslog message—the lower the number, the more severe the condition.
Message_number	A unique six-digit number that identifies the syslog message.
Message_text	A text string that describes the condition. This portion of the syslog message sometimes includes IP addresses, port numbers, or usernames.

Severity Levels

The following table lists the syslog message severity levels. You can assign custom colors to each of the severity levels to make it easier to distinguish them in the ASDM log viewers. To configure syslog message color settings, either choose the **Tools > Preferences > Syslog** tab or, in the log viewer itself, click **Color Settings** on the toolbar.

Table 47: Syslog Message Severity Levels

Level Number	Severity Level	Description
0	**emergencies**	System is unusable.
1	**alert**	Immediate action is needed.
2	**critical**	Critical conditions.
3	**error**	Error conditions.
4	**warning**	Warning conditions.
5	**notification**	Normal but significant conditions.
6	**informational**	Informational messages only.
7	**debugging**	Debugging messages only. Log at this level only temporarily, when debugging issues. This log level can potentially generate so many messages that system performance can be affected.

Note ASA does not generate syslog messages with a severity level of zero (emergencies).

Syslog Message Filtering

You can filter generated syslog messages so that only certain syslog messages are sent to a particular output destination. For example, you could configure the ASA to send all syslog messages to one output destination and to send a subset of those syslog messages to a different output destination.

Specifically, you can direct syslog messages to an output destination according to the following criteria:

- Syslog message ID number
- Syslog message severity level
- Syslog message class (equivalent to a functional area)

You customize these criteria by creating a message list that you can specify when you set the output destination. Alternatively, you can configure the ASA to send a particular message class to each type of output destination independently of the message list.

Syslog Message Classes

You can use syslog message classes in two ways:

- Specify an output location for an entire category of syslog messages. Use the **logging class** command.
- Create a message list that specifies the message class. Use the **logging list** command.

The syslog message class provides a method of categorizing syslog messages by type, equivalent to a feature or function of the device. For example, the rip class denotes RIP routing.

All syslog messages in a particular class share the same initial three digits in their syslog message ID numbers. For example, all syslog message IDs that begin with the digits 611 are associated with the vpnc (VPN client) class. Syslog messages associated with the VPN client feature range from 611101 to 611323.

In addition, most of the ISAKMP syslog messages have a common set of prepended objects to help identify the tunnel. These objects precede the descriptive text of a syslog message when available. If the object is not known at the time that the syslog message is generated, the specific heading = value combination does not appear.

The objects are prefixed as follows:

Group = *groupname*, Username = *user*, IP = *IP_address*

Where the group is the tunnel-group, the username is the username from the local database or AAA server, and the IP address is the public IP address of the remote access client or Layer 2 peer.

The following table lists the message classes and the range of message IDs in each class.

Table 48: Syslog Message Classes and Associated Message ID Numbers

Class	Definition	Syslog Message ID Numbers
auth	User Authentication	109, 113
—	Access Lists	106
—	Application Firewall	415

Class	Definition	Syslog Message ID Numbers
bridge	Transparent Firewall	110, 220
ca	PKI Certification Authority	717
citrix	Citrix Client	723
—	Clustering	747
—	Card Management	323
config	Command Interface	111, 112, 208, 308
csd	Secure Desktop	724
cts	Cisco TrustSec	776
dap	Dynamic Access Policies	734
eap, eapoudp	EAP or EAPoUDP for Network Admission Control	333, 334
eigrp	EIGRP Routing	336
email	E-mail Proxy	719
—	Environment Monitoring	735
ha	Failover	101, 102, 103, 104, 105, 210, 311, 709
—	Identity-based Firewall	746
ids	Intrusion Detection System	400, 733
—	IKEv2 Toolkit	750, 751, 752
ip	IP Stack	209, 215, 313, 317, 408
ipaa	IP Address Assignment	735
ips	Intrusion Protection System	400, 401, 420
—	IPv6	325
—	Botnet traffic filtering.	338
—	Licensing	444
mdm-proxy	MDM Proxy	802
nac	Network Admission Control	731, 732
nacpolicy	NAC Policy	731
nacsettings	NAC Settings to apply NAC Policy	732
—	Network Access Point	713

Class	Definition	Syslog Message ID Numbers
np	Network Processor	319
—	NP SSL	725
ospf	OSPF Routing	318, 409, 503, 613
—	Password Encryption	742
—	Phone Proxy	337
rip	RIP Routing	107, 312
rm	Resource Manager	321
—	Smart Call Home	120
session	User Session	106, 108, 201, 202, 204, 302, 303, 304, 305, 314, 405, 406, 407, 500, 502, 607, 608, 609, 616, 620, 703, 710
snmp	SNMP	212
—	ScanSafe	775
ssl	SSL Stack	725
svc	SSL VPN Client	722
sys	System	199, 211, 214, 216, 306, 307, 315, 414, 604, 605, 606, 610, 612, 614, 615,701, 711, 741
—	Threat Detection	733
tre	Transactional Rule Engine	780
—	UC-IME	339
tag-switching	Service Tag Switching	779
vm	VLAN Mapping	730
vpdn	PPTP and L2TP Sessions	213, 403, 603
vpn	IKE and IPsec	316, 320, 402, 404, 501, 602, 702, 713, 714, 715
vpnc	VPN Client	611
vpnfo	VPN Failover	720
vpnlb	VPN Load Balancing	718
—	VXLAN	778

Class	Definition	Syslog Message ID Numbers
webfo	WebVPN Failover	721
webvpn	WebVPN and AnyConnect Client	716
—	NAT and PAT	305

Custom Message Lists

Creating a custom message list is a flexible way to exercise control over which syslog messages are sent to which output destination. In a custom syslog message list, you specify groups of syslog messages using any or all of the following criteria:

- Severity level
- Message IDs
- Ranges of syslog message IDs
- Message class.

For example, you can use message lists to do the following:

- Select syslog messages with the severity levels of 1 and 2 and send them to one or more e-mail addresses.
- Select all syslog messages associated with a message class (such as ha) and save them to the internal buffer.

A message list can include multiple criteria for selecting messages. However, you must add each message selection criterion with a new command entry. It is possible to create a message list that includes overlapping message selection criteria. If two criteria in a message list select the same message, the message is logged only once.

Clustering

Syslog messages are an invaluable tool for accounting, monitoring, and troubleshooting in a clustering environment. Each ASA unit in the cluster (up to eight units are allowed) generates syslog messages independently; certain **logging** commands then enable you to control header fields, which include a time stamp and device ID. The syslog server uses the device ID to identify the syslog generator. You can use the **logging device-id** command to generate syslog messages with identical or different device IDs to make messages appear to come from the same or different units in the cluster.

Guidelines for Logging

This section includes guidelines and limitations that you should review before configuring logging.

IPv6 Guidelines

Does not support IPv6.

Additional Guidelines

- The syslog server must run a server program called syslogd. Windows provides a syslog server as part of its operating system.
- To view logs generated by the ASA, you must specify a logging output destination. If you enable logging without specifying a logging output destination, the ASA generates messages but does not save them to a location from which you can view them. You must specify each different logging output destination separately. For example, to designate more than one syslog server as an output destination, enter a new command for each syslog server.
- Sending syslogs over TCP is not supported on a standby device.
- It is not possible to have two different lists or classes being assigned to different syslog servers or same locations.
- You can configure up to 16 syslog servers. However, in multiple context mode, the limitation is 4 servers per context.
- The syslog server should be reachable through the ASA. You should configure the device to deny ICMP unreachable messages on the interface through which the syslog server is reachable and to send syslogs to the same server. Make sure that you have enabled logging for all severity levels. To prevent the syslog server from crashing, suppress the generation of syslogs 313001, 313004, and 313005.
- When you use a custom message list to match only access list hits, the access list logs are not generated for access lists that have had their logging severity level increased to debugging (level 7). The default logging severity level is set to 6 for the **logging list** command. This default behavior is by design. When you explicitly change the logging severity level of the access list configuration to debugging, you must also change the logging configuration itself.

  The following is sample output from the **show running-config logging** command that does not include access list hits, because their logging severity level has been changed to debugging:

  ```
  ciscoasa# show running-config logging
  logging enable
  logging timestamp
  logging list test message 106100
  logging buffered test
  ```

  The following is sample output from the **show running-config logging** command that does include access list hits:

  ```
  ciscoasa# show running-config logging
  logging enable
  logging timestamp
  logging buffered debugging
  ```

  In this case, the access list configuration does not change and the number of access list hits appears, as shown in the following example:

  ```
  ciscoasa(config)# access-list global line 1 extended
  permit icmp any host 4.2.2.2 log debugging interval 1 (hitcnt=7) 0xf36b5386
  ciscoasa(config)# access-list global line 2 extended
  permit tcp host 10.1.1.2 any eq www log informational interval 1 (hitcnt=18) 0xe7e7c3b8
  ciscoasa(config)# access-list global line 3 extended
  ```

```
permit ip any any (hitcnt=543) 0x25f9e609
```

- When the ASA sends syslogs via TCP, the connection takes about one minute to initiate after the syslogd service restarts.

Configure Logging

This section describes how to configure logging.

Enable Logging

To enable logging, perform the following steps:

Procedure

Enable logging.

logging enable

Example:

```
ciscoasa(config)# logging enable
```

Configure an Output Destination

To optimize syslog message usage for troubleshooting and performance monitoring, we recommend that you specify one or more locations where syslog messages should be sent, including an internal log buffer, one or more external syslog servers, ASDM, an SNMP management station, the console port, specified e-mail addresses, or Telnet and SSH sessions.

When you configure syslog logging on an interface with management-only access enabled, the dataplane related logs (syslog IDs 302015, 302014, 106023, and 304001) are dropped and does not reach the syslog server. The syslog messages are dropped because the datapath routing table does not have the management interface routing. Hence, ensure the interface that you are configuring has management-only access disabled

Send Syslog Messages to an External Syslog Server

You can archive messages according to the available disk space on the external syslog server, and manipulate logging data after it is saved. For example, you could specify actions to be executed when certain types of syslog messages are logged, extract data from the log and save the records to another file for reporting, or track statistics using a site-specific script.

To send syslog messages to an external syslog server, perform the following steps:

Procedure

Step 1 Configure the ASA to send messages to syslog servers.

logging host *interface_name syslog_ip* [**tcp**[/*port*] | **udp** [/*port*] [**format emblem**]]

Example:

```
ciscoasa(config)# logging host dmz1 192.168.1.5 udp/1026
```

The **format emblem** keyword enables EMBLEM format logging for the syslog server with UDP only. The *interface_name* argument specifies the interface through which you access the syslog server. The *syslog_ip* argument specifies the IP address of the syslog server. The **tcp**[/*port*] or **udp**[/*port*] keyword-argument pair specify that the ASA should use TCP or UDP to send syslog messages to the syslog server.

You can configure the ASA to send data to a syslog server using either UDP or TCP, but not both. The default protocol is UDP if you do not specify a protocol.

Warning If you specify TCP, when the ASA discovers syslog server failures, for security reasons, new connections through the ASA are blocked. To allow new connections regardless of connectivity to a TCP syslog server, see Step 3.

If you specify UDP, the ASA continues to allow new connections whether or not the syslog server is operational. Valid port values for either protocol are 1025 through 65535. The default UDP port is 514. The default TCP port is 1470.

Step 2 Specify which syslog messages should be sent to the syslog server.

logging trap {*severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging trap errors
```

You can specify the severity level number (1 through 7) or name. For example, if you set the severity level to 3, then the ASA sends syslog messages for severity levels 3, 2, and 1. You can specify a custom message list that identifies the syslog messages to send to the syslog server.

Step 3 (Optional) Disable the feature to block new connections when a TCP-connected syslog server is down.

logging permit-hostdown

Example:

```
ciscoasa(config)# logging permit-hostdown
```

When the ASA is configured to send syslog messages to a TCP-based syslog server, and if either the syslog server is down or the log queue is full, then new connections to ASA are blocked. New connections are allowed again after the syslog server is back up and the log queue is no longer full. Using this command, you can permit new connections even if the syslog server is not operational.

Step 4 (Optional) Set the logging facility to a value other than 20, which is what most UNIX systems expect.

logging facility *number*

Example:

```
ciscoasa(config)# logging facility 21
```

Enable Secure Logging

Procedure

Enable secure logging by specifying the **secure** keyword in the logging host command.

logging host *interface_name syslog_ip* [**tcp**/*port* | **udp**/*port*] [**format emblem**] [**secure**]

Where:

- **logging host** *interface_name syslog_ip* specifies the interface on which the syslog server resides and the IP address of the syslog server.
- [**tcp**/*port* | **udp**/*port*] specifies the port (TCP or UDP) that the syslog server listens to for syslog messages. The **tcp** keyword specifies that the ASA should use TCP to send syslog messages to the syslog server. The **udp** keyword specifies that the ASA should use UDP to send syslog messages to the syslog server.
- **format emblem** keyword enables EMBLEM format logging for the syslog server.
- **secure** keyword specifies that the connection to the remote logging host should use SSL/TLS for TCP only. Secure logging does not support UDP; an error occurs if you try to use this protocol.

Example:

```
ciscoasa(config)# logging host inside 10.0.0.1 TCP/1500 secure
```

Generate Syslog Messages in EMBLEM Format to a Syslog Server

To generate syslog messages in EMBLEM format to a syslog server, perform the following steps:

Procedure

Send syslog messages in EMBLEM format to a syslog server over UDP using port 514.

logging host *interface_name ip_address*{**tcp** [/*port*] | **udp** [/ *port*]] [**format emblem**]

Example:

```
ciscoasa(config)# logging host interface_1 127.0.0.1 udp format emblem
ciscoasa(config)# logging host interface_1 2001::1 udp format emblem
```

The **format emblem** keyword enables EMBLEM format logging for the syslog server (UDP only). The *interface_name* argument specifies the interface through which you access the syslog server. The *ip_address* argument specifies the IP address of the syslog server. The **tcp**[/*port*] or **udp**[/*port*] keyword and argument pair specify that the ASA should use TCP or UDP to send syslog messages to the syslog server.

You can configure the ASA to send data to a syslog server using either UDP or TCP. The default protocol is UDP if you do not specify a protocol.

You can use multiple **logging host** commands to specify additional servers that would all receive syslog messages. If you configure two or more logging servers, make sure that you limit the logging severity level to warnings for all logging servers.

Warning If you specify TCP, when the ASA discovers syslog server failures, for security reasons, new connections through the ASA are blocked. To permit new connections despite syslog server failures, see Step 3 of Send Syslog Messages to an External Syslog Server, on page 965.

If you specify UDP, the ASA continues to allow new connections whether or not the syslog server is operational. Valid port values for either protocol are 1025 through 65535. The default UDP port is 514. The default TCP port is 1470.

Generate Syslog Messages in EMBLEM Format to Other Output Destinations

To generate syslog messages in EMBLEM format to other output destinations, perform the following steps:

Procedure

Send syslog messages in EMBLEM format to output destinations other than a syslog server, such as Telnet or SSH sessions.

logging emblem

Example:

```
ciscoasa(config)# logging emblem
```

Send Syslog Messages to the Internal Log Buffer

You need to specify which syslog messages should be sent to the internal log buffer, which serves as a temporary storage location. New messages are appended to the end of the list. When the buffer is full, that is, when the buffer wraps, old messages are overwritten as new messages are generated, unless you configure the ASA to save the full buffer to another location.

To send syslog messages to the internal log buffer, perform the following steps:

Procedure

Step 1 Specify which syslog messages should be sent to the internal log buffer, which serves as a temporary storage location.

logging buffered {*severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging buffered critical
ciscoasa(config)# logging buffered level 2
ciscoasa(config)# logging buffered notif-list
```

New messages are appended to the end of the list. When the buffer is full, that is, when the buffer wraps, old messages are overwritten as new messages are generated, unless you configure the ASA to save the full buffer to another location. To empty the internal log buffer, enter the **clear logging buffer** command.

Step 2 Change the size of the internal log buffer. The default buffer size is 4 KB.

logging buffer-size *bytes*

Example:

```
ciscoasa(config)# logging buffer-size 16384
```

Step 3 Choose one of the following options:

- Save new messages to the internal log buffer and save the full log buffer content to the internal flash memory.

  logging flash-bufferwrap

  Example:

  ```
  ciscoasa(config)# logging flash-bufferwrap
  ```

- Save new messages to the internal log buffer and save the full log buffer content to an FTP server.

  logging ftp-bufferwrap

  Example:

  ```
  ciscoasa(config)# logging flash-bufferwrap
  ```

  When saving the buffer content to another location, the ASA create log files with names that use the following time-stamp format:

  ```
  LOG-YYYY-MM-DD-HHMMSS.TXT
  ```

  where *YYYY* is the year, *MM* is the month, *DD* is the day of the month, and *HHMMSS* is the time in hours, minutes, and seconds.

- Identify the FTP server on which you want to store log buffer content.

  logging ftp-server *server pathusername password*

  Example:

  ```
  ciscoasa(config)# logging ftp-server 10.1.1.1 /syslogs logsupervisor 1luvMy10gs
  ```

  The *server* argument specifies the IP address of the external FTP server. The *path* argument specifies the directory path on the FTP server where the log buffer data is to be saved. This path is relative to the

FTP root directory. The *username* argument specifies a username that is valid for logging into the FTP server. The *password* argument indicates the password for the username specified.

- Save the current log buffer content to the internal flash memory.

  logging savelog [*savefile*]

  Example:

  ```
  ciscoasa(config)# logging savelog latest-logfile.txt
  ```

Change the Amount of Internal Flash Memory Available for Logs

To change the amount of internal flash memory available for logs, perform the following steps:

Procedure

Step 1 Specify the maximum amount of internal flash memory available for saving log files.

logging flash-maximum-allocation *kbytes*

Example:

```
ciscoasa(config)# logging flash-maximum-allocation 1200
```

By default, the ASA can use up to 1 MB of internal flash memory for log data. The minimum amount of internal flash memory that must be free for the ASA to save log data is 3 MB.

If a log file being saved to internal flash memory would cause the amount of free internal flash memory to fall below the configured minimum limit, the ASA deletes the oldest log files to ensure that the minimum amount of memory remains free after saving the new log file. If there are no files to delete or if, after all old files have been deleted, free memory is still below the limit, the ASA fails to save the new log file.

Step 2 Specify the minimum amount of internal flash memory that must be free for the ASA to save a log file.

logging flash-minimum-free *kbytes*

Example:

```
ciscoasa(config)# logging flash-minimum-free 4000
```

Send Syslog Messages to an E-mail Address

To send syslog messages to an e-mail address, perform the following steps:

Procedure

Step 1 Specify which syslog messages should be sent to an e-mail address.

logging mail {*severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging mail high-priority
```

When sent by e-mail, a syslog message appears in the subject line of the e-mail message. For this reason, we recommend configuring this option to notify administrators of syslog messages with high severity levels, such as critical, alert, and emergency.

Step 2 Specify the source e-mail address to be used when sending syslog messages to an e-mail address.

logging from-address *email_address*

Example:

```
ciscoasa(config)# logging from-address xxx-001@example.com
```

Step 3 Specify the recipient e-mail address to be used when sending syslog messages to an e-mail address.

logging recipient-address *e-mail_address*[*severity_level*]

Example:

```
ciscoasa(config)# logging recipient-address admin@example.com
```

Step 4 Specify the SMTP server to be used when sending syslog messages to an e-mail address.

Example:

```
ciscoasa(config)# smtp-server 10.1.1.24
```

Send Syslog Messages to ASDM

To send syslog messages to ASDM, perform the following steps:

Procedure

Step 1 Specify which syslog messages should be sent to ASDM.

logging asdm {*severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging asdm 2
```

The ASA sets aside a buffer area for syslog messages waiting to be sent to ASDM and saves messages in the buffer as they occur. The ASDM log buffer is a different buffer than the internal log buffer. When the ASDM log buffer is full, the ASA deletes the oldest syslog message to make room in the buffer for new ones. Deletion of the oldest syslog message to make room for new ones is the default setting in ASDM. To control the number of syslog messages retained in the ASDM log buffer, you can change the size of the buffer.

Step 2 Specify the number of syslog messages to be retained in the ASDM log buffer.

logging asdm-buffer-size *num_of_msgs*

Example:

```
ciscoasa(config)# logging asdm-buffer-size 200
```

Enter the **clear logging asdm** command to empty the current content of the ASDM log buffer.

Configure the Logging Queue

To configure the logging queue, perform the following steps:

Procedure

Specify the number of syslog messages that the ASA can hold in its queue before sending them to the configured output destination.

logging queue *message_count*

Example:

```
ciscoasa(config)# logging queue 300
```

The ASA have a fixed number of blocks in memory that can be allocated for buffering syslog messages while they are waiting to be sent to the configured output destination. The number of blocks required depends on the length of the syslog message queue and the number of syslog servers specified. The default queue size is 512 syslog messages. The queue size is limited only by block memory availability. Valid values are from 0 to 8192 messages, depending on the platform. If the logging queue is set to zero, the queue is the maximum configurable size (8192 messages).

Send Syslog Messages to the Console Port

To send syslog messages to the console port, perform the following steps:

Procedure

Specify which syslog messages should be sent to the console port.

logging console { *severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging console errors
```

Send Syslog Messages to an SNMP Server

To enable logging to an SNMP server, perform the following steps:

Procedure

Enable SNMP logging and specify which messages are to be sent to SNMP servers.

logging history [*logging_list* | *level*]

Example:

```
ciscoasa(config)# logging history errors
```

Enter the **no logging history** command to disable SNMP logging.

Send Syslog Messages to a Telnet or SSH Session

To send syslog messages to a Telnet or SSH session, perform the following steps:

Procedure

Step 1 Specify which syslog messages should be sent to a Telnet or SSH session.

logging monitor {*severity_level* | *message_list*}

Example:

```
ciscoasa(config)# logging monitor 6
```

Step 2 Enable logging to the current session only.

terminal monitor

Example:

```
ciscoasa(config)# terminal monitor
```

If you log out and then log in again, you need to reenter this command. Enter the **terminal no monitor** command to disable logging to the current session.

Configure Syslog Messages

Show or Hide Invalid Usernames in Syslogs

You can show or hide invalid usernames in syslog messages for unsuccessful login attempts. The default setting is to hide usernames when the username is invalid or if the validity is unknown. If a user accidentally types a password instead of a username, for example, then it is more secure to hide the "username" in the resultant syslog message. You might want to show invalid usernames to help with troubleshooting login issues.

Procedure

Step 1 Show invalid usernames:

no logging hide username

Step 2 Hide invalid usernames:

logging hide username

Include the Date and Time in Syslog Messages

To include the date and time in syslog messages, perform the following steps:

Procedure

Specify that syslog messages should include the date and time that they were generated.

logging timestamp

Example:

```
ciscoasa(config)# logging timestamp
LOG-2008-10-24-081856.TXT
```

To remove the date and time from syslog messages, enter the **no logging timestamp** command.

Disable a Syslog Message

To disable a specified syslog message, perform the following steps:

Procedure

Prevent the ASA from generating a particular syslog message.

no logging message *syslog_id*

Example:

```
ciscoasa(config)# no logging message 113019
```

To reenable a disabled syslog message, enter the **logging message** *syslog_id* command (for example, **logging message 113019**). To reenable logging of all disabled syslog messages, enter the **clear configure logging disabled** command.

Change the Severity Level of a Syslog Message

To change the severity level of a syslog message, perform the following steps:

Procedure

Specify the severity level of a syslog message.

logging message *syslog_id* **level** *severity_level*

Example:

```
ciscoasa(config)# logging message 113019 level 5
```

To reset the severity level of a syslog message to its setting, enter the **no logging message** *syslog_id* **level** *severity_level* command (for example, **no logging message 113019 level 5**). To reset the severity level of all modified syslog messages to their settings, enter the **clear configure logging level** command.

Block Syslog Messages on a Standby Unit

Procedure

Use the following command to block a specific syslog message from being generated on a standby unit.

no logging message *syslog-id* **standby**

Example:

```
ciscoasa(config)# no logging message 403503 standby
```

Unblock a specific syslog message to ensure that the syslog messages of the failover standby ASA stay synchronized if failover occurs. Use the **logging standby** command to unblock a specific syslog message that was previously blocked from being generated on a standby unit.

Note During a steady state when both active and standby ASAs are logging, the traffic doubles on the shared logging destinations, such as syslog servers, SNMP servers, and FTP servers. However, at times of a failover, during the switchover phase, the standby ASA generates more events including switchover intrusion and connection events of the active unit.

Include the Device ID in Non-EMBLEM Format Syslog Messages

To include the device ID in non-EMBLEM format syslog messages, perform the following steps:

Procedure

Configure the ASA to include a device ID in non-EMBLEM-format syslog messages. You can specify only one type of device ID for syslog messages.

logging device-id {**cluster-id** | **context-name** | **hostname** | **ipaddress** *interface_name* [**system**] | **string** *text*}

Example:

```
ciscoasa(config)# logging device-id hostname
ciscoasa(config)# logging device-id context-name
```

The **context-name** keyword indicates that the name of the current context should be used as the device ID (applies to multiple context mode only). If you enable the logging device ID for the admin context in multiple context mode, messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID.

Note In an ASA cluster, always use the control unit IP address for the selected interface.

The **cluster-id** keyword specifies the unique name in the boot configuration of an individual ASA unit in the cluster as the device ID. The **hostname** keyword specifies that the hostname of the ASA should be used as the device ID. The **ipaddress** *interface_name* keyword-argument pair specifies that the interface IP address specified as *interface_name* should be used as the device ID. If you use the **ipaddress** keyword, the device ID becomes the specified ASA interface IP address, regardless of the interface from which the syslog message is sent. In the cluster environment, the **system** keyword dictates that the device ID becomes the system IP address on the interface. This keyword provides a single, consistent device ID for all syslog messages that are sent from the device. The **string** *text* keyword-argument pair specifies that the text string should be used as the device ID. The string can include as many as 16 characters.

You cannot use blank spaces or any of the following characters:

- & (ampersand)
- ‘ (single quote)

- “ (double quote)
- < (less than)
- > (greater than)
- ? (question mark)

Note If enabled, the device ID does not appear in EMBLEM-formatted syslog messages nor in SNMP traps.

Create a Custom Event List

You use the following three criteria to define an event list:

- Event Class
- Severity
- Message ID

To create a custom event list to send to a specific logging destination (for example, an SNMP server), perform the following steps:

Procedure

Step 1 Specify criteria for selecting messages to be saved in the internal log buffer. For example, if you set the severity level to 3, then the ASA sends syslog messages for severity levels 3, 2, and 1.

logging list *name* {**level** *level* [**class** *message_class*] | **message** *start_id*[-*end_id*]}

Example:

```
ciscoasa(config)# logging list list-notif level 3
```

The *name* argument specifies the name of the list. The **level** *level* keyword and argument pair specify the severity level. The **class** *message_class* keyword-argument pair specify a particular message class. The **message** *start_id* [-*end_id*] keyword-argument pair specify an individual syslog message number or a range of numbers.

Note Do not use the names of severity levels as the name of a syslog message list. Prohibited names include emergencies, alert, critical, error, warning, notification, informational, and debugging. Similarly, do not use the first three characters of these words at the beginning of an event list name. For example, do not use an event list name that starts with the characters “err.”

Step 2 (Optional) Add more criteria for message selection to the list.

logging list *name* {**level** *level* [**class** *message_class*] | **message** *start_id*[-*end_id*]}

Example:

```
ciscoasa(config)# logging list list-notif message 104024-105999
ciscoasa(config)# logging list list-notif level critical
ciscoasa(config)# logging list list-notif level warning class ha
```

Enter the same command as in the previous step, specifying the name of the existing message list and the additional criterion. Enter a new command for each criterion that you want to add to the list. For example, you can specify criteria for syslog messages to be included in the list as the following:

- Syslog message IDs that fall into the range of 104024 to 105999.
- All syslog messages with the critical severity level or higher (emergency, alert, or critical).
- All ha class syslog messages with the warning severity level or higher (emergency, alert, critical, error, or warning).

Note A syslog message is logged if it satisfies any of these conditions. If a syslog message satisfies more than one of the conditions, the message is logged only once.

Configure Logging Filters

Send All Syslog Messages in a Class to a Specified Output Destination

To send all syslog messages in a class to a specified output destination, perform the following steps:

Procedure

Override the configuration in the specified output destination command. For example, if you specify that messages at severity level 7 should go to the internal log buffer and that ha class messages at severity level 3 should go to the internal log buffer, then the latter configuration takes precedence.

logging class *message_class* {**buffered** | **console** | **history** | **mail** | **monitor** | **trap**} [*severity_level*]

Example:

```
ciscoasa(config)# logging class ha buffered alerts
```

The **buffered**, **history**, **mail**, **monitor**, and **trap** keywords specify the output destination to which syslog messages in this class should be sent. The **history** keyword enables SNMP logging. The **monitor** keyword enables Telnet and SSH logging. The **trap** keyword enables syslog server logging. Select one destination per command line entry. To specify that a class should go to more than one destination, enter a new command for each output destination.

Limit the Rate of Syslog Message Generation

To limit the rate of syslog message generation, perform the following steps:

Procedure

Apply a specified severity level (1 through 7) to a set of messages or to an individual message (not the destination) within a specified time period.

logging rate-limit {**unlimited** | {*num* [*interval*]}} **message** *syslog_id* | **level** *severity_level*

Example:

```
ciscoasa(config)# logging rate-limit 1000 600 level 6
```

Rate limits affect the volume of messages being sent to all configured destinations. To reset the logging rate limit to the default value, enter the **clear running-config logging rate-limit** command. To reset the logging rate limit, enter the **clear configure logging rate-limit** command.

Monitoring the Logs

See the following commands for monitoring logging status.

- **show logging**

  This command shows syslog messages, including the severity level.

  > **Note** The maximum number of syslog messages that are available to view is 1000, which is the default setting. The maximum number of syslog messages that are available to view is 2000.

- **show logging message**

  This command shows a list of syslog messages with modified severity levels and disabled syslog messages.

- **show logging message** *message_ID*

  This command shows the severity level of a specific syslog message.

- **show logging queue**

  This command shows the logging queue and queue statistics.

- **show running-config logging rate-limit**

  This command shows the current logging rate-limit setting.

Examples for Logging

The following examples show the logging information, that displays for the **show logging** command:

```
ciscoasa(config)# show logging
```

```
Syslog logging: enabled
    Facility: 16
    Timestamp logging: disabled
    Standby logging: disabled
    Deny Conn when Queue Full: disabled
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: level errors, facility 16, 3607 messages logged
        Logging to infrastructure 10.1.2.3
    History logging: disabled
    Device ID: 'inside' interface IP address “10.1.1.1”
    Mail logging: disabled
    ASDM logging: disabled
```

The following examples show how to control both whether a syslog message is enabled and the severity level of the specified syslog message:

```
ciscoasa(config)# show logging message 403503
syslog 403503: -level errors (enabled)

ciscoasa(config)# logging message 403503 level 1
ciscoasa(config)# show logging message 403503
syslog 403503: -level errors, current-level alerts (enabled)

ciscoasa(config)# no logging message 403503
ciscoasa(config)# show logging message 403503
syslog 403503: -level errors, current-level alerts (disabled)

ciscoasa(config)# logging message 403503
ciscoasa(config)# show logging message 403503
syslog 403503: -level errors, current-level alerts (enabled)

ciscoasa(config)# no logging message 403503 level 3
ciscoasa(config)# show logging message 403503
syslog 403503: -level errors (enabled)
```

History for Logging

Table 49: History for Logging

Feature Name	Platform Releases	Description
Logging	7.0(1)	Provides ASA network logging information through various output destinations, and includes the option to view and save log files.
Rate limit	7.0(4)	Limits the rate at which syslog messages are generated. We introduced the following command: **logging rate-limit**.

Feature Name	Platform Releases	Description
Logging list	7.2(1)	Creates a logging list to use in other commands to specify messages by various criteria (logging level, event class, and message IDs). We introduced the following command: **logging list**.
Secure logging	8.0(2)	Specifies that the connection to the remote logging host should use SSL/TLS. This option is valid only if the protocol selected is TCP. We modified the following command: **logging host**.
Logging class	8.0(4), 8.1(1)	Added support for the ipaa event class of logging messages. We modified the following command: **logging class**.
Logging class and saved logging buffers	8.2(1)	Added support for the dap event class of logging messages. We modified the following command: **logging class**. Added support to clear the saved logging buffers (ASDM, internal, FTP, and flash). We introduced the following command: **clear logging queue bufferwrap**.
Password encryption	8.3(1)	Added support for password encryption. We modified the following command: **logging ftp server**.
Log viewers	8.3(1)	The source and destination IP addresses were added to the log viewers.

Feature Name	Platform Releases	Description
Enhanced logging and connection blocking	8.3(2)	When you configure a syslog server to use TCP, and the syslog server is unavailable, the ASA blocks new connections that generate syslog messages until the server becomes available again (for example, VPN, firewall, and cut-through-proxy connections). This feature has been enhanced to also block new connections when the logging queue on the ASA is full; connections resume when the logging queue is cleared. This feature was added for compliance with Common Criteria EAL4+. Unless required, we recommended allowing connections when syslog messages cannot be sent or received. To allow connections, continue to use the **logging permit-hostdown** command. We introduced the following syslog messages: 414005, 414006, 414007, and 414008. We modified the following command: **show logging**.
Syslog message filtering and sorting	8.4(1)	Support has been added for the following: • Syslog message filtering based on multiple text strings that correspond to various columns • Creation of custom filters • Column sorting of messages. For detailed information, see the ASDM configuration guide. This feature interoperates with all ASA versions.
Clustering	9.0(1)	Added support for syslog message generation in a clustering environment on the ASA 5580 and 5585-X. We modified the following command: **logging device-id**.

Feature Name	Platform Releases	Description
Blocking syslogs on a standby unit	9.4(1)	We added support for blocking the generation of specific syslog messages on the standby unit in a failover configuration. We introduced the following command: **logging message** *syslog-id* **standby**.
IPv6 address support for syslog servers	9.7(1)	You can now configure syslog servers with IPv6 addresses to record, send, and receive syslogs over TCP and UDP. We modified the following command: **logging host**

CHAPTER 41

SNMP

This chapter describes how to configure Simple Network Management Protocol (SNMP) to monitor the Cisco ASA.

About SNMP

SNMP is an application-layer protocol that facilitates the exchange of management information between network devices and is part of the TCP/IP protocol suite. The ASA provides support for network monitoring using SNMP Versions 1, 2c, and 3, and support the use of all three versions simultaneously. The SNMP agent running on the ASA interface lets you monitor the network devices through network management systems (NMSes), such as HP OpenView. The ASA support SNMP read-only access through issuance of a GET request. SNMP write access is not allowed, so you cannot make changes with SNMP. In addition, the SNMP SET request is not supported.

You can configure the ASA to send traps, which are unsolicited messages from the managed device to the management station for certain events (event notifications) to an NMS, or you can use the NMS to browse the Management Information Bases (MIBs) on the security devices. MIBs are a collection of definitions, and the ASA maintain a database of values for each definition. Browsing a MIB means issuing a series of GET-NEXT or GET-BULK requests of the MIB tree from the NMS to determine values.

The ASA have an SNMP agent that notifies designated management stations if events occur that are predefined to require a notification, for example, when a link in the network goes up or down. The notification it sends includes an SNMP OID, which identifies itself to the management stations. The ASA agent also replies when a management station asks for information.

SNMP Terminology

The following table lists the terms that are commonly used when working with SNMP.

Table 50: SNMP Terminology

Term	Description
Agent	The SNMP server running on the ASA. The SNMP agent has the following features: • Responds to requests for information and actions from the network management station. • Controls access to its Management Information Base, the collection of objects that the SNMP manager can view or change. • Does not allow SET operations.
Browsing	Monitoring the health of a device from the network management station by polling required information from the SNMP agent on the device. This activity may include issuing a series of GET-NEXT or GET-BULK requests of the MIB tree from the network management station to determine values.
Management Information Bases (MIBs)	Standardized data structures for collecting information about packets, connections, buffers, failovers, and so on. MIBs are defined by the product, protocols, and hardware standards used by most network devices. SNMP network management stations can browse MIBs and request specific data or events be sent as they occur.
Network management stations (NMSs)	The PCs or workstations set up to monitor SNMP events and manage devices, such as the ASA.
Object identifier (OID)	The system that identifies a device to its NMS and indicates to users the source of information monitored and displayed.
Trap	Predefined events that generate a message from the SNMP agent to the NMS. Events include alarm conditions such as linkup, linkdown, coldstart, warmstart, authentication, or syslog messages.

MIBs and Traps

MIBs are either standard or enterprise-specific. Standard MIBs are created by the IETF and documented in various RFCs. A trap reports significant events occurring on a network device, most often errors or failures. SNMP traps are defined in either standard or enterprise-specific MIBs. Standard traps are created by the IETF and documented in various RFCs. SNMP traps are compiled into the ASA software.

If needed, you can also download RFCs, standard MIBs, and standard traps from the following locations:

http://www.ietf.org/

Browse the complete list of Cisco MIBs, traps, and OIDs from the following location:

ftp://ftp.cisco.com/pub/mibs/supportlists/asa/asa-supportlist.html

In addition, download Cisco OIDs by FTP from the following location:

ftp://ftp.cisco.com/pub/mibs/oid/oid.tar.gz

Note In software versions 7.2(1), 8.0(2), and later, the interface information accessed through SNMP refreshes about every 5 seconds. As a result, we recommend that you wait for at least 5 seconds between consecutive polls.

Not all OIDs in MIBs are supported. To obtain a list of the supported SNMP MIBs and OIDs for a specific ASA, enter the following command:

```
ciscoasa(config)# show snmp-server oidlist
```

Note Although the **oidlist** keyword does not appear in the options list for the **show snmp-server** command help, it is available. However, this command is for Cisco TAC use only. Contact the Cisco TAC before using this command.

The following is sample output from the **show snmp-server oidlist** command:

```
ciscoasa(config)# show snmp-server oidlist
[0]     1.3.6.1.2.1.1.1.        sysDescr
[1]     1.3.6.1.2.1.1.2.        sysObjectID
[2]     1.3.6.1.2.1.1.3.        sysUpTime
[3]     1.3.6.1.2.1.1.4.        sysContact
[4]     1.3.6.1.2.1.1.5.        sysName
[5]     1.3.6.1.2.1.1.6.        sysLocation
[6]     1.3.6.1.2.1.1.7.        sysServices
[7]     1.3.6.1.2.1.2.1.        ifNumber
[8]     1.3.6.1.2.1.2.2.1.1.    ifIndex
[9]     1.3.6.1.2.1.2.2.1.2.    ifDescr
[10]    1.3.6.1.2.1.2.2.1.3.    ifType
[11]    1.3.6.1.2.1.2.2.1.4.    ifMtu
[12]    1.3.6.1.2.1.2.2.1.5.    ifSpeed
[13]    1.3.6.1.2.1.2.2.1.6.    ifPhysAddress
[14]    1.3.6.1.2.1.2.2.1.7.    ifAdminStatus
[15]    1.3.6.1.2.1.2.2.1.8.    ifOperStatus
[16]    1.3.6.1.2.1.2.2.1.9.    ifLastChange
[17]    1.3.6.1.2.1.2.2.1.10.   ifInOctets
[18]    1.3.6.1.2.1.2.2.1.11.   ifInUcastPkts
[19]    1.3.6.1.2.1.2.2.1.12.   ifInNUcastPkts
[20]    1.3.6.1.2.1.2.2.1.13.   ifInDiscards
[21]    1.3.6.1.2.1.2.2.1.14.   ifInErrors
[22]    1.3.6.1.2.1.2.2.1.16.   ifOutOctets
[23]    1.3.6.1.2.1.2.2.1.17.   ifOutUcastPkts
[24]    1.3.6.1.2.1.2.2.1.18.   ifOutNUcastPkts
[25]    1.3.6.1.2.1.2.2.1.19.   ifOutDiscards
[26]    1.3.6.1.2.1.2.2.1.20.   ifOutErrors
[27]    1.3.6.1.2.1.2.2.1.21.   ifOutQLen
[28]    1.3.6.1.2.1.2.2.1.22.   ifSpecific
[29]    1.3.6.1.2.1.4.1.        ipForwarding
[30]    1.3.6.1.2.1.4.20.1.1.   ipAdEntAddr
[31]    1.3.6.1.2.1.4.20.1.2.   ipAdEntIfIndex
[32]    1.3.6.1.2.1.4.20.1.3.   ipAdEntNetMask
[33]    1.3.6.1.2.1.4.20.1.4.   ipAdEntBcastAddr
[34]    1.3.6.1.2.1.4.20.1.5.   ipAdEntReasmMaxSize
[35]    1.3.6.1.2.1.11.1.       snmpInPkts
[36]    1.3.6.1.2.1.11.2.       snmpOutPkts
[37]    1.3.6.1.2.1.11.3.       snmpInBadVersions
[38]    1.3.6.1.2.1.11.4.       snmpInBadCommunityNames
[39]    1.3.6.1.2.1.11.5.       snmpInBadCommunityUses
[40]    1.3.6.1.2.1.11.6.       snmpInASNParseErrs
[41]    1.3.6.1.2.1.11.8.       snmpInTooBigs
[42]    1.3.6.1.2.1.11.9.       snmpInNoSuchNames
[43]    1.3.6.1.2.1.11.10.      snmpInBadValues
[44]    1.3.6.1.2.1.11.11.      snmpInReadOnlys
[45]    1.3.6.1.2.1.11.12.      snmpInGenErrs
```

```
[46]    1.3.6.1.2.1.11.13.      snmpInTotalReqVars
[47]    1.3.6.1.2.1.11.14.      snmpInTotalSetVars
[48]    1.3.6.1.2.1.11.15.      snmpInGetRequests
[49]    1.3.6.1.2.1.11.16.      snmpInGetNexts
[50]    1.3.6.1.2.1.11.17.      snmpInSetRequests
[51]    1.3.6.1.2.1.11.18.      snmpInGetResponses
[52]    1.3.6.1.2.1.11.19.      snmpInTraps
[53]    1.3.6.1.2.1.11.20.      snmpOutTooBigs
[54]    1.3.6.1.2.1.11.21.      snmpOutNoSuchNames
[55]    1.3.6.1.2.1.11.22.      snmpOutBadValues
[56]    1.3.6.1.2.1.11.24.      snmpOutGenErrs
[57]    1.3.6.1.2.1.11.25.      snmpOutGetRequests
[58]    1.3.6.1.2.1.11.26.      snmpOutGetNexts
[59]    1.3.6.1.2.1.11.27.      snmpOutSetRequests
[60]    1.3.6.1.2.1.11.28.      snmpOutGetResponses
[61]    1.3.6.1.2.1.11.29.      snmpOutTraps
[62]    1.3.6.1.2.1.11.30.      snmpEnableAuthenTraps
[63]    1.3.6.1.2.1.11.31.      snmpSilentDrops
[64]    1.3.6.1.2.1.11.32.      snmpProxyDrops
[65]    1.3.6.1.2.1.31.1.1.1.1. ifName
[66]    1.3.6.1.2.1.31.1.1.1.2. ifInMulticastPkts
[67]    1.3.6.1.2.1.31.1.1.1.3. ifInBroadcastPkts
[68]    1.3.6.1.2.1.31.1.1.1.4. ifOutMulticastPkts
[69]    1.3.6.1.2.1.31.1.1.1.5. ifOutBroadcastPkts
[70]    1.3.6.1.2.1.31.1.1.1.6. ifHCInOctets
--More--
```

SNMP Object Identifiers

Each Cisco system-level product has an SNMP object identifier (OID) for use as a MIB-II sysObjectID. The CISCO-PRODUCTS-MIB and the CISCO-ENTITY-VENDORTYPE-OID-MIB includes the OIDs that can be reported in the sysObjectID object in the SNMPv2-MIB, Entity Sensor MIB and Entity Sensor Threshold Ext MIB. You can use this value to identify the model type. The following table lists the sysObjectID OIDs for ASA and ISA models.

Table 51: SNMP Object Identifiers

Product Identifier	sysObjectID	Model Number
ASA 5506 Adaptive Security Appliance	ciscoASA5506 (ciscoProducts 2114)	ASA 5506-X
ASA 5506 Adaptive Security Appliance Security Context	ciscoASA5506sc (ciscoProducts 2115)	ASA 5506-X security context
ASA 5506 Adaptive Security Appliance System Context	ciscoASA5506sy (ciscoProducts 2116)	ASA 5506-X system context
ASA 5506W Adaptive Security Appliance	ciscoASA5506W (ciscoProducts 2117)	ASA 5506W-X
ASA 5506W Adaptive Security Appliance Security Context	ciscoASA5506Wsc (ciscoProducts 2118)	ASA 5506W-X security context
ASA 5506W Adaptive Security Appliance System Context	ciscoASA5506Wsy (ciscoProducts 2119)	ASA 5506W-X system context
ASA 5508 Adaptive Security Appliance	ciscoASA5508 (ciscoProducts 2120)	ASA 5508-X

Product Identifier	sysObjectID	Model Number
ASA 5508 Adaptive Security Appliance Security Context	ciscoASA5508sc (ciscoProducts 2121)	ASA 5508-X security context
ASA 5508 Adaptive Security Appliance System Context	ciscoASA5508sy (ciscoProducts 2122)	ASA 5508-X system context
ASA 5506 Adaptive Security Appliance with No Payload Encryption	ciscoASA5506K7 (ciscoProducts 2123)	ASA 5506-X Adaptive Security Appliance with No Payload Encryption
ASA 5506 Adaptive Security Appliance Security Context with No Payload Encryption	ciscoASA5506K7sc (ciscoProducts 2124)	ASA 5506-X Adaptive Security Appliance Security Context with No Payload Encryption
ASA 5506 Adaptive Security Appliance System Context with No Payload Encryption	ciscoASA5506K7sy (ciscoProducts 2125)	ASA 5506-X Adaptive Security Appliance System Context with No Payload Encryption
ASA 5508 Adaptive Security Appliance with No Payload Encryption	ciscoASA5508K7 (ciscoProducts 2126)	ASA 5508-X Adaptive Security Appliance System Context with No Payload Encryption
ASA 5508 Adaptive Security Appliance Security Context with No Payload Encryption	ciscoASA5508K7sc (ciscoProducts 2127)	ASA 5508-X Adaptive Security Appliance Security Context with No Payload Encryption
ASA 5508 Adaptive Security Appliance System Context with No Payload Encryption	ciscoASA5508K7sy (ciscoProducts 2128)	ASA 5508-X Adaptive Security Appliance System Context with No Payload Encryption
ASA5585-SSP10	ciscoASA5585Ssp10 (ciscoProducts 1194)	ASA 5585-X SSP-10
ASA5585-SSP20	ciscoASA5585Ssp20 (ciscoProducts 1195)	ASA 5585-X SSP-20
ASA5585-SSP40	ciscoASA5585Ssp40 (ciscoProducts 1196)	ASA 5585-X SSP-40
ASA5585-SSP60	ciscoASA5585Ssp60 (ciscoProducts 1197)	ASA 5585-X SSP-60
ASA5585-SSP10	ciscoASA5585Ssp10sc (ciscoProducts 1198)	ASA 5585-X SSP-10 security context
ASA5585-SSP20	ciscoASA5585Ssp20sc (ciscoProducts 1199)	ASA 5585-X SSP-20 security context
ASA5585-SSP40	ciscoASA5585Ssp40sc (ciscoProducts 1200)	ASA 5585-X SSP-40 security context
ASA5585-SSP60	ciscoASA5585Ssp60sc (ciscoProducts 1201)	ASA 5585-X SSP-60 security context
ASA5585-SSP10	ciscoASA5585Ssp10sy (ciscoProducts 1202)	ASA 5585-X SSP-10 system context

Product Identifier	sysObjectID	Model Number
ASA5585-SSP20	ciscoASA5585Ssp20sy (ciscoProducts 1203)	ASA 5585-X SSP-20 system context
ASA5585-SSP40	ciscoASA5585Ssp40sy (ciscoProducts 1204)	ASA 5585-X SSP-40 system context
ASA5585-SSP60	ciscoASA5585Ssp60sy (ciscoProducts 1205)	ASA 5585-X SSP-60 system context
ASA Services Module for Catalyst switches/7600 routers	ciscoAsaSm1 (ciscoProducts 1277)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers
ASA Services Module for Catalyst switches/7600 routers security context	ciscoAsaSm1sc (ciscoProducts 1275)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers security context
ASA Services Module for Catalyst switches/7600 routers security context with No Payload Encryption	ciscoAsaSm1K7sc (ciscoProducts 1334)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers security context with No Payload Encryption
ASA Services Module for Catalyst switches/7600 routers system context	ciscoAsaSm1sy (ciscoProducts 1276)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers system context
ASA Services Module for Catalyst switches system context/7600 routers with No Payload Encryption	ciscoAsaSm1K7sy (ciscoProducts 1335)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers system context with No Payload Encryption
ASA Services Module for Catalyst switches/7600 routers system context with No Payload Encryption	ciscoAsaSm1K7 (ciscoProducts 1336)	Adaptive Security Appliance (ASA) Services Module for Catalyst switches/7600 routers with No Payload Encryption
ASA 5512	ciscoASA5512 (ciscoProducts 1407)	ASA 5512 Adaptive Security Appliance
ASA 5525	ciscoASA5525 (ciscoProducts 1408)	ASA 5525 Adaptive Security Appliance
ASA 5545	ciscoASA5545 (ciscoProducts 1409)	ASA 5545 Adaptive Security Appliance
ASA 5555	ciscoASA5555 (ciscoProducts 1410)	ASA 5555 Adaptive Security Appliance
ASA 5512 Security Context	ciscoASA5512sc (ciscoProducts 1411)	ASA 5512 Adaptive Security Appliance Security Context
ASA 5525 Security Context	ciscoASA5525sc (ciscoProducts 1412)	ASA 5525 Adaptive Security Appliance Security Context
ASA 5545 Security Context	ciscoASA5545sc (ciscoProducts 1413)	ASA 5545 Adaptive Security Appliance Security Context

Product Identifier	sysObjectID	Model Number
ASA 5555 Security Context	ciscoASA5555sc (ciscoProducts 1414)	ASA 5555 Adaptive Security Appliance Security Context
ASA 5512 System Context	ciscoASA5512sy (ciscoProducts 1415)	ASA 5512 Adaptive Security Appliance System Context
ASA 5515 System Context	ciscoASA5515sy (ciscoProducts 1416)	ASA 5515 Adaptive Security Appliance System Context
ASA 5525 System Context	ciscoASA5525sy (ciscoProducts1417)	ASA 5525 Adaptive Security Appliance System Context
ASA 5545 System Context	ciscoASA5545sy (ciscoProducts 1418)	ASA 5545 Adaptive Security Appliance System Context
ASA 5555 System Context	ciscoASA5555sy (ciscoProducts 1419)	ASA 5555 Adaptive Security Appliance System Context
ASA 5515 Security Context	ciscoASA5515sc (ciscoProducts 1420)	ASA 5515 Adaptive Security Appliance System Context
ASA 5515	ciscoASA5515 (ciscoProducts 1421)	ASA 5515 Adaptive Security Appliance
ASAv	ciscoASAv (ciscoProducts 1902)	Cisco Adaptive Security Virtual Appliance (ASAv)
ASAv System Context	ciscoASAvsy (ciscoProducts 1903)	Cisco Adaptive Security Virtual Appliance (ASAv) System Context
ASAv Security Context	ciscoASAvsc (ciscoProducts 1904)	Cisco Adaptive Security Virtual Appliance (ASAv) Security Contex
ISA 30004C Industrial Security Appliance	ciscoProducts 2268	ciscoISA30004C
CISCO ISA30004C with 4 GE Copper Security Context	ciscoProducts 2139	ciscoISA30004Csc
CISCO ISA30004C with 4 GE Copper System Context	ciscoProducts 2140	ciscoISA30004Csy
ISA 30002C2F Industrial Security Appliance	ciscoProducts 2267	ciscoISA30002C2F
CISCO ISA30002C2F with 2 GE Copper ports + 2 GE Fiber Security Context	ciscoProducts 2142	ciscoISA30002C2Fsc
CISCO ISA30002C2F with 2 GE Copper ports + 2 GE Fiber System Context	ciscoProducts 2143	ciscoISA30002C2Fsy
Cisco Industrial Security Appliance (ISA) 30004C Chassis	cevChassis 1677	cevChassisISA30004C

Product Identifier	sysObjectID	Model Number
Cisco Industrial Security Appliance (ISA) 30002C2F Chassis	cevChassis 1678	cevChassisISA30002C2F
Central Processing Unit Temperature Sensor for ISA30004C Copper SKU	cevSensor 187	cevSensorISA30004CCpuTempSensor
Central Processing Unit Temperature Sensor for ISA30002C2F Fiber	cevSensor 189	cevSensorISA30002C2FCpuTempSensor
Processor Card Temperature Sensor for ISA30004C Copper SKU	cevSensor 192	cevSensorISA30004CPTS
Processor Card Temperature Sensor for ISA30002C2F Fiber SKU	cevSensor 193	cevSensorISA30002C2FPTS
Power Card Temperature Sensor for ISA30004C Copper SKU	cevSensor 197	cevSensorISA30004CPowercardTS
Power Card Temperature Sensor for ISA30002C2F Fiber SKU	cevSensor 198	cevSensorISA30002C2FPowercardTS
Port Card Temperature Sensor for ISA30004C	cevSensor 199	cevSensorISA30004CPortcardTS
Port Card Temperature Sensor for ISA30002C2F	cevSensor 200	cevSensorISA30002C2FPortcardTS
Central Processing Unit for ISA30004C Copper SKU	cevModuleCpuType 329	cevCpuISA30004C
Central Processing Unit for ISA30002C2F Fiber SKU	cevModuleCpuType 330	cevCpuISA30002C2F
Modules ISA30004C, ISA30002C2F	cevModule 111	cevModuleISA3000Type
30004C Industrial Security Appliance Solid State Drive	cevModuleISA3000Type 1	cevModuleISA30004CSSD64
30002C2F Industrial Security Appliance Solid State Drive	cevModuleISA3000Type 2	cevModuleISA30002C2FSSD64
Cisco ISA30004C/ISA30002C2F Hardware Bypass	cevModuleISA3000Type 5	cevModuleISA3000HardwareBypass

Physical Vendor Type Values

Each Cisco chassis or standalone system has a unique type number for SNMP use. The entPhysicalVendorType OIDs are defined in the CISCO-ENTITY-VENDORTYPE-OID-MIB. This value is returned in the entPhysicalVendorType object from the ASA, ASAv, or ASASM SNMP agent. You can use this value to identify the type of component (module, power supply, fan, sensors, CPU, and so on). The following table lists the physical vendor type values for the ASA models.

Table 52: Physical Vendor Type Values

Item	entPhysicalVendorType OID Description
ASA Services Module for Catalyst switches/7600 routers	cevCat6kWsSvcAsaSm1 (cevModuleCat6000Type 169)
ASA Services Module for Catalyst switches/7600 routers with No Payload Encryption	cevCat6kWsSvcAsaSm1K7 (cevModuleCat6000Type 186)
Accelerator for 5506 Adaptive Security Appliance	cevAcceleratorAsa5506 (cevOther 10)
Accelerator for 5506W Adaptive Security Appliance	cevAcceleratorAsa5506W (cevOther 11)
Accelerator for 5508 Adaptive Security Appliance	cevAcceleratorAsa5508 (cevOther 12)
Accelerator for 5506 with No Payload Encryption Adaptive Security Appliance	cevAcceleratorAsa5506K7 (cevOther 13)
Accelerator for 5508 with No Payload Encryption Adaptive Security Appliance	cevAcceleratorAsa5508K7 (cevOther 14)
Cisco Adaptive Security Appliance (ASA) 5506 Chassis	cevChassisAsa5506 (cevChassis 1600)
Cisco Adaptive Security Appliance (ASA) 5506W Chassis	cevChassisAsa5506W (cevChassis 1601)
Cisco Adaptive Security Appliance (ASA) 5508 Chassis	cevChassisAsa5508 (cevChassis 1602)
Cisco Adaptive Security Appliance (ASA) 5506 Chassis with No Payload Encryption	cevChassisAsa5506K7 (cevChassis 1603)
Cisco Adaptive Security Appliance (ASA) 5508 Chassis with No Payload Encryption	cevChassisAsa5508K7 (cevChassis 1604)
Central Processing Unit for 5506 Adaptive Security Appliance	cevCpuAsa5506 (cevModuleCpuType 312)
Central Processing Unit for 5506W Adaptive Security Appliance	cevCpuAsa5506W (cevModuleCpuType 313)
Central Processing Unit for 5508 Adaptive Security Appliance	cevCpuAsa5508 ((cevModuleCpuType 314)
Central Processing Unit for 5506 with No Payload Encryption Adaptive Security Appliance	cevCpuAsa5506K7 (cevModuleCpuType 315)
Central Processing Unit for 5508 with No Payload Encryption Adaptive Security Appliance	cevCpuAsa5508K7 (cevModuleCpuType 316)
cevModuleASA5506 Type chassis	cevModuleASA5506Type (cevModule 107)
5506 Adaptive Security Appliance Field-Replaceable Solid State Drive	cevModuleAsa5506SSD (cevModuleASA5506Type 1)
5506W Adaptive Security Appliance Field-Replaceable Solid State Drive	cevModuleAsa5506WSSD (cevModuleASA5506Type 2)
5506 with No Payload Encryption Adaptive Security Appliance Field-Replaceable Solid State Drive	cevModuleAsa5506K7SSD (cevModuleASA5506Type 3)

Item	entPhysicalVendorType OID Description
cevModuleASA5508 Type chassis	cevModuleASA5508Type (cevModule 108)
5508 Adaptive Security Appliance Field-Replaceable Solid State Drive	cevModuleAsa5508SSD (cevModuleASA5508Type 1)
5508 with No Payload Encryption Adaptive Security Appliance Field-Replaceable Solid State Drive	cevModuleAsa5508K7SSD (cevModuleASA5508Type 2)
Chassis Cooling Fan for Adaptive Security Appliance 5508	cevFanAsa5508ChassisFan (cevFan 247)
Chassis Cooling Fan for Adaptive Security Appliance 5508 with No Payload Encryption	cevFanAsa5508K7ChassisFan (cevFan 248)
Chassis Cooling Fan Sensor for Adaptive Security Appliance 5508	cevSensorAsa5508ChassisFanSensor (cevSensor 162)
Chassis Cooling Fan Sensor for Adaptive Security Appliance 5508 with No Payload Encryption	cevSensorAsa5508K7ChassisFanSensor (cevSensor 163)
Central Processing Unit Temperature Sensor for 5506 Adaptive Security Appliance	cevSensorAsa5506CpuTempSensor (cevSensor 164)
Central Processing Unit Temperature Sensor for 5506W Adaptive Security Appliance	cevSensorAsa5506WCpuTempSensor (cevSensor 165)
Central Processing Unit Temperature Sensor for 5508 Adaptive Security Appliance	cevSensorAsa5508CpuTempSensor (cevSensor 166)
Central Processing Unit Temperature Sensor for 5506 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5506K7CpuTempSensor (cevSensor 167)
Central Processing Unit Temperature Sensor for 5508 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5508K7CpuTempSensor (cevSensor 168)
Accelerator Temperature Sensor for 5506 Adaptive Security Appliance	cevSensorAsa5506AcceleratorTempSensor (cevSensor 169)
Accelerator Temperature Sensor for 5506W Adaptive Security Appliance	cevSensorAsa5506WAcceleratorTempSensor (cevSensor 170)
Accelerator Temperature Sensor for 5508 Adaptive Security Appliance	cevSensorAsa5508AcceleratorTempSensor (cevSensor 171)
Accelerator Temperature Sensor for 5506 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5506K7AcceleratorTempSensor (cevSensor 172)
Accelerator Temperature Sensor for 5508 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5508K7AcceleratorTempSensor (cevSensor 173)
Chassis Ambient Temperature Sensor for 5506 Adaptive Security Appliance	cevSensorAsa5506ChassisTempSensor (cevSensor 174)

Item	entPhysicalVendorType OID Description
Chassis Ambient Temperature Sensor for 5506W Adaptive Security Appliance	cevSensorAsa5506WChassisTempSensor (cevSensor 175)
Chassis Ambient Temperature Sensor for 5508 Adaptive Security Appliance	cevSensorAsa5508ChassisTempSensor (cevSensor 176)
Chassis Ambient Temperature Sensor for 5506 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5506K7ChassisTempSensor (cevSensor 177)
Chassis Ambient Temperature Sensor for 5508 with No Payload Encryption Adaptive Security Appliance	cevSensorAsa5508K7ChassisTempSensor (cevSensor 178)
Cisco Adaptive Security Appliance (ASA) 5512 Adaptive Security Appliance	cevChassisASA5512 (cevChassis 1113)
Cisco Adaptive Security Appliance (ASA) 5512 Adaptive Security Appliance with No Payload Encryption	cevChassisASA5512K7 (cevChassis 1108)
Cisco Adaptive Security Appliance (ASA) 5515 Adaptive Security Appliance	cevChassisASA5515 (cevChassis 1114)
Cisco Adaptive Security Appliance (ASA) 5515 Adaptive Security Appliance with No Payload Encryption	cevChassisASA5515K7 (cevChassis 1109)
Cisco Adaptive Security Appliance (ASA) 5525 Adaptive Security Appliance	cevChassisASA5525 (cevChassis 1115)
Cisco Adaptive Security Appliance (ASA) 5525 Adaptive Security Appliance with No Payload Encryption	cevChassisASA5525K7 (cevChassis 1110)
Cisco Adaptive Security Appliance (ASA) 5545 Adaptive Security Appliance	cevChassisASA5545 (cevChassis 1116)
Cisco Adaptive Security Appliance (ASA) 5545 Adaptive Security Appliance with No Payload Encryption	cevChassisASA5545K7 (cevChassis 1111)
Cisco Adaptive Security Appliance (ASA) 5555 Adaptive Security Appliance	cevChassisASA5555 (cevChassis 1117)
Cisco Adaptive Security Appliance (ASA) 5555 Adaptive Security Appliance with No Payload Encryption	cevChassisASA5555K7 (cevChassis 1112)
Central Processing Unit for Cisco Adaptive Security Appliance 5512	cevCpuAsa5512 (cevModuleCpuType 229)
Central Processing Unit for Cisco Adaptive Security Appliance 5512 with no Payload Encryption	cevCpuAsa5512K7 (cevModuleCpuType 224)
Central Processing Unit for Cisco Adaptive Security Appliance 5515	cevCpuAsa5515 (cevModuleCpuType 230)
Central Processing Unit for Cisco Adaptive Security Appliance 5515 with no Payload Encryption	cevCpuAsa5515K7 (cevModuleCpuType 225)

Item	entPhysicalVendorType OID Description
Central Processing Unit for Cisco Adaptive Security Appliance 5525	cevCpuAsa5525 (cevModuleCpuType 231)
Central Processing Unit for Cisco Adaptive Security Appliance 5525 with no Payload Encryption	cevCpuAsa5525K7 (cevModuleCpuType 226)
Central Processing Unit for Cisco Adaptive Security Appliance 5545	cevCpuAsa5545 (cevModuleCpuType 232)
Central Processing Unit for Cisco Adaptive Security Appliance 5545 with no Payload Encryption	cevCpuAsa5545K7 (cevModuleCpuType 227)
Central Processing Unit for Cisco Adaptive Security Appliance 5555	cevCpuAsa5555 (cevModuleCpuType 233)
Central Processing Unit for Cisco Adaptive Security Appliance 5555 with no Payload Encryption	cevCpuAsa5555K7 (cevModuleCpuType 228)
CPU for ASA 5585 SSP-10	cevCpuAsa5585Ssp10 (cevModuleCpuType 204)
CPU for ASA 5585 SSP-10 No Payload Encryption	cevCpuAsa5585Ssp10K7 (cevModuleCpuType 205)
CPU for ASA 5585 SSP-20	cevCpuAsa5585Ssp20 (cevModuleCpuType 206)
CPU for ASA 5585 SSP-20 No Payload Encryption	cevCpuAsa5585Ssp20K7 (cevModuleCpuType 207)
CPU for ASA 5585 SSP-40	cevCpuAsa5585Ssp40 (cevModuleCpuType 208)
CPU for ASA 5585 SSP-40 No Payload Encryption	cevCpuAsa5585Ssp40K7 (cevModuleCpuType 209)
CPU for ASA 5585 SSP-60	cevCpuAsa5585Ssp60 (cevModuleCpuType 210)
CPU for ASA 5585 SSP-60 No Payload Encryption	cevCpuAsa5585Ssp60K (cevModuleCpuType 211)
CPU for Cisco ASA Services Module for Catalyst switches/7600 routers	cevCpuAsaSm1 (cevModuleCpuType 222)
CPU for Cisco ASA Services Module with No Payload Encryption for Catalyst switches/7600 routers	cevCpuAsaSm1K7 (cevModuleCpuType 223)
Chassis Cooling Fan in Adaptive Security Appliance 5512	cevFanASA5512ChassisFan (cevFan 163)
Chassis Cooling Fan in Adaptive Security Appliance 5512 with No Payload Encryption	cevFanASA5512K7ChassisFan (cevFan 172)
Chassis Cooling Fan in Adaptive Security Appliance 5515	cevFanASA5515ChassisFan (cevFan 164)
Chassis Cooling Fan in Adaptive Security Appliance 5515 with No Payload Encryption	cevFanASA5515K7ChassisFan (cevFan 171)
Chassis Cooling Fan in Adaptive Security Appliance 5525	cevFanASA5525ChassisFan (cevFan 165)
Chassis Cooling Fan in Adaptive Security Appliance 5525 with No Payload Encryption	cevFanASA5525K7ChassisFan (cevFan 170)

Item	entPhysicalVendorType OID Description
Chassis Cooling Fan in Adaptive Security Appliance 5545	cevFanASA5545ChassisFan (cevFan 166)
Chassis Cooling Fan in Adaptive Security Appliance 5545 with No Payload Encryption	cevFanASA5545K7ChassisFan (cevFan 169)
Power Supply Fan in Adaptive Security Appliance 5545 with No Payload Encryption	cevFanASA5545K7PSFan (cevFan 161)
Power Supply Fan in Adapative Security Appliance 5545	cevFanASA5545PSFan (cevFan 159)
Chassis Cooling Fan in Adaptive Security Appliance 5555	cevFanASA5555ChassisFan (cevFan 167)
Chassis Cooling Fan in Adaptive Security Appliance 5555 with No Payload Encryption	cevFanASA5555K7ChassisFan (cevFan 168)
Power Supply Fan in Adaptive Security Appliance 5555	cevFanASA5555PSFan (cevFan 160)
Power Supply Fan in Adaptive Security Appliance 5555 with No Payload Encryption	cevFanASA5555PSFanK7 (cevFan 162)
Power supply fan for ASA 5585-X	cevFanASA5585PSFan (cevFan 146)
10-Gigabit Ethernet interface	cevPort10GigEthernet (cevPort 315)
Gigabit Ethernet port	cevPortGe (cevPort 109)
Power Supply unit in Adaptive Security Appliance 5545	cevPowerSupplyASA5545PSInput (cevPowerSupply 323)
Presence Sensor for Power Supply input in Adaptive Security Appliance 5545	cevPowerSupplyASA5545PSPresence (cevPowerSupply 321)
Power Supply unit in Adaptive Security Appliance 5555	cevPowerSupplyASA5555PSInput (cevPowerSupply 324)
Presence Sensor for Power Supply input in Adaptive Security Appliance 5555	cevPowerSupplyASA5555PSPresence (cevPowerSupply 322)
Power supply input for ASA 5585	cevPowerSupplyASA5585PSInput (cevPowerSupply 304)
Cisco Adaptive Security Appliance (ASA) 5512 Chassis Fan sensor	cevSensorASA5512ChassisFanSensor (cevSensor 120)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5512	cevSensorASA5512ChassisTemp (cevSensor 107)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5512	cevSensorASA5512CPUTemp (cevSensor 96)
Cisco Adaptive Security Appliance (ASA) 5512 with No Payload Encryption Chassis Fan sensor	cevSensorASA5512K7ChassisFanSensor (cevSensor 125)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5512 with No Payload Encryption	cevSensorASA5512K7CPUTemp (cevSensor 102)

Item	entPhysicalVendorType OID Description
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5512 with No Payload Encryption	cevSensorASA5512K7PSFanSensor (cevSensor 116)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5512	cevSensorASA5512PSFanSensor (cevSensor 119)
Cisco Adaptive Security Appliance (ASA) 5515 Chassis Fan sensor	cevSensorASA5515ChassisFanSensor (cevSensor 121)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5515	cevSensorASA5515ChassisTemp (cevSensor 98)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5515	cevSensorASA5515CPUTemp (cevSensor 97)
Cisco Adaptive Security Appliance (ASA) 5515 with No Payload Encryption Chassis Fan sensor	cevSensorASA5515K7ChassisFanSensor (cevSensor 126)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5515 with No Payload Encryption	cevSensorASA5515K7CPUTemp (cevSensor 103)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5515 with No Payload Encryption	cevSensorASA5515K7PSFanSensor (cevSensor 115)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5515	cevSensorASA5515PSFanSensor (cevSensor 118)
Cisco Adaptive Security Appliance (ASA) 5525 Chassis Fan sensor	cevSensorASA5525ChassisFanSensor (cevSensor 122)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5525	cevSensorASA5525ChassisTemp (cevSensor 108)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5525	cevSensorASA5525CPUTemp (cevSensor 99)
Cisco Adaptive Security Appliance (ASA) 5525 with No Payload Encryption Chassis Fan sensor	cevSensorASA5525K7ChassisFanSensor (cevSensor 127)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5525 with No Payload Encryption	cevSensorASA5525K7CPUTemp (cevSensor 104)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5525 with No Payload Encryption	cevSensorASA5525K7PSFanSensor (cevSensor 114)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5525	cevSensorASA5525PSFanSensor (cevSensor 117)
Cisco Adaptive Security Appliance (ASA) 5545 Chassis Fan sensor	cevSensorASA5545ChassisFanSensor (cevSensor 123)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5545	cevSensorASA5545ChassisTemp (cevSensor 109)

Item	entPhysicalVendorType OID Description
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5545	cevSensorASA5545CPUTemp (cevSensor 100)
Cisco Adaptive Security Appliance (ASA) 5545 with No Payload Encryption Chassis Fan sensor	cevSensorASA5545K7ChassisFanSensor (cevSensor 128)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545K7ChassisTemp (cevSensor 90)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545K7CPUTemp (cevSensor 105)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545K7PSFanSensor (cevSensor 113)
Presence Sensor for Power Supply input in Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545K7PSPresence (cevSensor 87)
Temperature Sensor for Power Supply Fan in Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545K7PSTempSensor (cevSensor 94)
Sensor for Power Supply Fan in Adaptive Security Appliance 5545 with No Payload Encryption	cevSensorASA5545PSFanSensor (cevSensor 89)
Presence Sensor for Power Supply input in Adaptive Security Appliance 5545	cevSensorASA5545PSPresence (cevSensor 130)
Presence Sensor for Power Supply input in Adaptive Security Appliance 5555	cevSensorASA5545PSPresence (cevSensor 131)
Temperature Sensor for Power Supply Fan in Adaptive Security Appliance 5545	cevSensorASA5545PSTempSensor (cevSensor 92)
Cisco Adaptive Security Appliance (ASA) 5555 Chassis Fan sensor	cevSensorASA5555ChassisFanSensor (cevSensor 124)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5555	cevSensorASA5555ChassisTemp (cevSensor 110)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5555	cevSensorASA5555CPUTemp (cevSensor 101)
Cisco Adaptive Security Appliance (ASA) 5555 with No Payload Encryption Chassis Fan sensor	cevSensorASA5555K7ChassisFanSensor (cevSensor 129)
Chassis Ambient Temperature Sensor for Cisco Adaptive Security Appliance 5555 with No Payload Encryption	cevSensorASA5555K7ChassisTemp (cevSensor 111)
Central Processing Unit Temperature Sensor for Cisco Adaptive Security Appliance 5555 with No Payload Encryption	cevSensorASA5555K7CPUTemp (cevSensor 106)
Sensor for Chassis Cooling Fan in Adaptive Security Appliance 5555 with No Payload Encryption	cevSensorASA5555K7PSFanSensor (cevSensor 112)

Item	entPhysicalVendorType OID Description
Presence Sensor for Power Supply input in Adaptive Security Appliance 5555 with No Payload Encryption	cevSensorASA5555K7PSPresence (cevSensor 88)
Temperature Sensor for Power Supply Fan in Adaptive Security Appliance 5555 with No Payload Encryption	cevSensorASA5555K7PSTempSensor (cevSensor 95)
Sensor for Power Supply Fan in Adaptive Security Appliance 5555	cevSensorASA5555PSFanSensor (cevSensor 91)
Temperature Sensor for Power Supply Fan in Adaptive Security Appliance 5555	cevSensorASA5555PSTempSensor (cevSensor 93)
Sensor for power supply fan for ASA 5585-X	cevSensorASA5585PSFanSensor (cevSensor 86)
Sensor for power supply input for ASA 5585-X	cevSensorASA5585PSInput (cevSensor 85)
CPU temperature sensor for ASA 5585 SSP-10	cevSensorASA5585SSp10CPUTemp (cevSensor 77)
CPU temperature sensor for ASA 5585 SSP-10 No Payload Encryption	cevSensorASA5585SSp10K7CPUTemp (cevSensor 78)
CPU temperature sensor for ASA 5585 SSP-20	cevSensorASA5585SSp20CPUTemp (cevSensor 79)
CPU temperature sensor for ASA 5585 SSP-20 No Payload Encryption	cevSensorASA5585SSp20K7CPUTemp (cevSensor 80)
CPU temperature sensor for ASA 5585 SSP-40	cevSensorASA5585SSp40CPUTemp (cevSensor 81)
CPU temperature sensor for ASA 5585 SSP-40 No Payload Encryption	cevSensorASA5585SSp40K7CPUTemp (cevSensor 82)
CPU temperature sensor for ASA 5585 SSP-60	cevSensorASA5585SSp60CPUTemp (cevSensor 83)
CPU temperature sensor for ASA 5585 SSP-60 No Payload Encryption	cevSensorASA5585SSp60K7CPUTemp (cevSensor 84)
Adaptive Security Appliance 5555-X Field-Replaceable Solid State Drive	cevModuleASA5555XFRSSD (cevModuleCommonCards 396)
Adaptive Security Appliance 5545-X Field-Replaceable Solid State Drive	cevModuleASA5545XFRSSD (cevModuleCommonCards 397)
Adaptive Security Appliance 5525-X Field-Replaceable Solid State Drive	cevModuleASA5525XFRSSD (cevModuleCommonCards 398)
Adaptive Security Appliance 5515-X Field-Replaceable Solid State Drive	cevModuleASA5515XFRSSD (cevModuleCommonCards 399)
Adaptive Security Appliance 5512-X Field-Replaceable Solid State Drive	cevModuleASA5512XFRSSD (cevModuleCommonCards 400)
Cisco Adaptive Security Virtual Appliance	cevChassisASAv (cevChassis 1451)

Supported Tables and Objects in MIBs

The following table lists the supported tables and objects for the specified MIBs.

In multi-context mode, these tables and objects provide information for a single context. If you want data across contexts, you need to sum them. For example, to get overall memory usage, sum the cempMemPoolHCUsed values for each context.

Table 53: Supported Tables and Objects in MIBs

MIB Name	Supported Tables and Objects
CISCO-ENHANCED-MEMPOOL-MIB	cempMemPoolTable, cempMemPoolIndex, cempMemPoolType, cempMemPoolName, cempMemPoolAlternate, cempMemPoolValid, cempMemPoolUsed, cempMemPoolFree, cempMemPoolUsedOvrflw, cempMemPoolHCUsed, cempMemPoolFreeOvrflw, cempMemPoolHCFree
CISCO-ENTITY-SENSOR-EXT-MIB **Note** Not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.	ceSensorExtThresholdTable
CISCO-L4L7MODULE-RESOURCE-LIMIT-MIB	ciscoL4L7ResourceLimitTable
CISCO-TRUSTSEC-SXP-MIB **Note** Not supported on the Cisco Adaptive Security Virtual Appliance (ASAv).	ctsxSxpGlobalObjects, ctsxSxpConnectionObjects, ctsxSxpSgtObjects
DISMAN-EVENT-MIB	mteTriggerTable, mteTriggerThresholdTable, mteObjectsTable, mteEventTable, mteEventNotificationTable
DISMAN-EXPRESSION-MIB **Note** Not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.	expExpressionTable, expObjectTable, expValueTable
ENTITY-SENSOR-MIB **Note** Not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers. **Note** Provides information related to physical sensors, such as chassis temperature, fan RPM, power supply voltage, etc. Not supported on the Cisco ASAv platform.	entPhySensorTable
NAT-MIB	natAddrMapTable, natAddrMapIndex, natAddrMapName, natAddrMapGlobalAddrType, natAddrMapGlobalAddrFrom, natAddrMapGlobalAddrTo, natAddrMapGlobalPortFrom, natAddrMapGlobalPortTo, natAddrMapProtocol, natAddrMapAddrUsed, natAddrMapRowStatus

MIB Name	Supported Tables and Objects
CISCO-PTP-MIB **Note** Only MIBs corresponding to E2E Transparent Clock mode are supported.	ciscoPtpMIBSystemInfo, cPtpClockDefaultDSTable, cPtpClockTransDefaultDSTable, cPtpClockPortTransDSTable

Supported Traps (Notifications)

The following table lists the supported traps (notifications) and their associated MIBs.

Table 54: Supported Traps (Notifications)

Trap and MIB Name	Varbind List	Description
authenticationFailure (SNMPv2-MIB)	—	For SNMP Version 1 or 2, the community string provided in the SNMP request is incorrect. For SNMP Version 3, a report PDU is generated instead of a trap if the auth or priv passwords or usernames are incorrect. The **snmp-server enable traps snmp authentication** command is used to enable and disable transmission of these traps.
ccmCLIRunningConfigChanged (CISCO-CONFIG-MAN-MIB)	—	The **snmp-server enable traps config** command is used to enable transmission of this trap.
cefcFRUInserted (CISCO-ENTITY-FRU-CONTROL -MIB)	—	The **snmp-server enable traps entity fru-insert** command is used to enable this notification. This trap does not apply to the ASA 5506-X and ASA 5508-X.
cefcFRURemoved (CISCO-ENTITY-FRU-CONTROL -MIB)	—	The **snmp-server enable traps entity fru-remove** command is used to enable this notification. This trap does not apply to the ASA 5506-X and ASA 5508-X.

Trap and MIB Name	Varbind List	Description
ceSensorExtThresholdNotification (CISCO-ENTITY-SENSOR-EXT -MIB) **Note** Not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.	entPhysicalName, entPhysicalDescr, entPhySensorValue, entPhySensorType, ceSensorExtThresholdValue	

Trap and MIB Name	Varbind List	Description
		The **snmp-server enable traps entity** [**power-supply-failure** | **fan-failure** | **cpu-temperature**] command is used to enable transmission of the entity threshold notifications. This notification is sent for a power supply failure. The objects sent identify the fan and CPU temperature. The **snmp-server enable traps entity fan-failure** command is used to enable transmission of the fan failure trap. This trap does not apply to the ASA 5506-X and ASA 5508-X. The **snmp-server enable traps entity power-supply-failure** command is used to enable transmission of the power supply failure trap. This trap does not apply to the ASA 5506-X and ASA 5508-X. The **snmp-server enable traps entity chassis-fan-failure** command is used to enable transmission of the chassis fan failure trap. This trap does not apply to the ASA 5506-X and ASA 5508-X. The **snmp-server enable traps entity cpu-temperature** command is used to enable transmission of the high CPU temperature trap. The **snmp-server enable traps entity power-supply-presence** command is used to enable transmission of the power supply presence failure trap. This trap does not apply to the ASA 5506-X and ASA 5508-X. The **snmp-server enable traps entity power-supply-temperature** command is used to enable transmission of the power supply temperature threshold trap. This trap does not apply to the ASA 5506-X and ASA 5508-X. The **snmp-server enable traps entity chassis-temperature** command is used to enable transmission of the chassis ambient temperature trap. The **snmp-server enable traps entity accelerator-temperature** command is used to enable transmission of the chassis

Trap and MIB Name	Varbind List	Description
		accelerator temperature trap. This trap does not apply to the ASA 5506-X and ASA 5508-X.
cipSecTunnelStart (CISCO-IPSEC-FLOW-MONITOR -MIB)	cipSecTunLifeTime, cipSecTunLifeSize	The **snmp-server enable traps ipsec start** command is used to enable transmission of this trap.
cipSecTunnelStop (CISCO-IPSEC-FLOW-MONITOR -MIB)	cipSecTunActiveTime	The **snmp-server enable traps ipsec stop** command is used to enable transmission of this trap.
ciscoConfigManEvent (CISCO-CONFIG-MAN-MIB)	—	The **snmp-server enable traps config** command is used to enable transmission of this trap.
ciscoRasTooManySessions (CISCO-REMOTE-ACCESS -MONITOR-MIB)	crasNumSessions, crasNumUsers, crasMaxSessionsSupportable, crasMaxUsersSupportable, crasThrMaxSessions	The **snmp-server enable traps remote-access session-threshold-exceeded** command is used to enable transmission of these traps.
clogMessageGenerated (CISCO-SYSLOG-MIB)	clogHistFacility, clogHistSeverity, clogHistMsgName, clogHistMsgText, clogHistTimestamp	Syslog messages are generated. The value of the clogMaxSeverity object is used to decide which syslog messages are sent as traps. The **snmp-server enable traps syslog** command is used to enable and disable transmission of these traps.
clrResourceLimitReached (CISCO-L4L7MODULE-RESOURCE -LIMIT-MIB)	crlResourceLimitValueType, crlResourceLimitMax, clogOriginIDType, clogOriginID	The **snmp-server enable traps connection-limit-reached** command is used to enable transmission of the connection-limit-reached notification. The clogOriginID object includes the context name from which the trap originated.
coldStart (SNMPv2-MIB)	—	The SNMP agent has started. The **snmp-server enable traps snmp coldstart** command is used to enable and disable transmission of these traps.
cpmCPURisingThreshold (CISCO-PROCESS-MIB)	cpmCPURisingThresholdValue, cpmCPUTotalMonIntervalValue, cpmCPUInterruptMonIntervalValue, cpmCPURisingThresholdPeriod, cpmProcessTimeCreated, cpmProcExtUtil5SecRev	The **snmp-server enable traps cpu threshold rising** command is used to enable transmission of the CPU threshold rising notification. The cpmCPURisingThresholdPeriod object is sent with the other objects.

Trap and MIB Name	Varbind List	Description
entConfigChange (ENTITY-MIB)	—	The **snmp-server enable traps entity config-change fru-insert fru-remove** command is used to enable this notification. **Note** This notification is only sent in multimode when a security context is created or removed.
linkDown (IF-MIB)	ifIndex, ifAdminStatus, ifOperStatus	The linkdown trap for interfaces. The **snmp-server enable traps snmp linkdown** command is used to enable and disable transmission of these traps.
linkUp (IF-MIB)	ifIndex, ifAdminStatus, ifOperStatus	The linkup trap for interfaces. The **snmp-server enable traps snmp linkup** command is used to enable and disable transmission of these traps.
mteTriggerFired (DISMAN-EVENT-MIB)	mteHotTrigger, mteHotTargetName, mteHotContextName, mteHotOID, mteHotValue, cempMemPoolName, cempMemPoolHCUsed	The **snmp-server enable traps memory-threshold** command is used to enable the memory threshold notification. The mteHotOID is set to cempMemPoolHCUsed. The cempMemPoolName and cempMemPoolHCUsed objects are sent with the other objects.
mteTriggerFired (DISMAN-EVENT-MIB) **Note** Not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.	mteHotTrigger, mteHotTargetName, mteHotContextName, mteHotOID, mteHotValue, ifHCInOctets, ifHCOutOctets, ifHighSpeed, entPhysicalName	The **snmp-server enable traps interface-threshold** command is used to enable the interface threshold notification. The entPhysicalName objects are sent with the other objects.
natPacketDiscard (NAT-MIB)	ifIndex	The **snmp-server enable traps nat packet-discard** command is used to enable the NAT packet discard notification. This notification is rate limited for 5 minutes and is generated when IP packets are discarded by NAT because mapping space is not available. The ifIndex gives the ID of the mapped interface.
warmStart (SNMPv2-MIB)	—	The **snmp-server enable traps snmp warmstart** command is used to enable and disable transmission of these traps.

Interface Types and Examples

The interface types that produce SNMP traffic statistics include the following:

- Logical—Statistics collected by the software driver, which are a subset of physical statistics.
- Physical—Statistics collected by the hardware driver. Each physical named interface has a set of logical and physical statistics associated with it. Each physical interface may have more than one VLAN interface associated with it. VLAN interfaces only have logical statistics.

  Note For a physical interface that has multiple VLAN interfaces associated with it, be aware that SNMP counters for ifInOctets and ifOutoctets OIDs match the aggregate traffic counters for that physical interface.

- VLAN-only—SNMP uses logical statistics for ifInOctets and ifOutOctets.

The examples in the following table show the differences in SNMP traffic statistics. Example 1 shows the difference in physical and logical output statistics for the **show interface** command and the **show traffic** command. Example 2 shows output statistics for a VLAN-only interface for the **show interface** command and the **show traffic** command. The example shows that the statistics are close to the output that appears for the **show traffic** command.

Table 55: SNMP Traffic Statistics for Physical and VLAN Interfaces

<table>
<tr><th>Example 1</th><th>Example 2</th></tr>
<tr><td>

```
ciscoasa# show interface GigabitEthernet3/2
interface GigabitEthernet3/2
description fullt-mgmt
nameif mgmt
security-level 10
ip address 10.7.14.201 255.255.255.0
management-only

ciscoasa# show traffic
(Condensed output)

Physical Statistics
GigabitEthernet3/2:
received (in 121.760 secs)
36 packets       3428 bytes
0 pkts/sec      28 bytes/sec

Logical Statistics
mgmt:
received (in 117.780 secs)
36 packets       2780 bytes
0 pkts/sec      23 bytes/sec
```

The following examples show the SNMP output statistics for the management interface and the physical interface. The ifInOctets value is close to the physical statistics output that appears in the **show traffic** command output but not to the logical statistics output.

ifIndex of the mgmt interface:

```
IF_MIB::ifDescr.6 = Adaptive Security Appliance ‘mgmt’
 interface
```

ifInOctets that corresponds to the physical interface statistics:

```
IF-MIB::ifInOctets.6 = Counter32:3246
```

</td><td>

```
ciscoasa# show interface GigabitEthernet0/0.100
interface GigabitEthernet0/0.100
vlan 100
nameif inside
security-level 100
ip address 10.7.1.101 255.255.255.0 standby 10.7.1.102

ciscoasa# show traffic
inside
received (in 9921.450 secs)
1977 packets       126528 bytes
0 pkts/sec      12 bytes/sec
transmitted (in 9921.450 secs)
1978 packets       126556 bytes
0 pkts/sec      12 bytes/sec
```

ifIndex of VLAN inside:

```
IF-MIB::ifDescr.9 = Adaptive Security Appliance ‘inside’
 interface
IF-MIB::ifInOctets.9 = Counter32: 126318
```

</td></tr>
</table>

SNMP Version 3 Overview

SNMP Version 3 provides security enhancements that are not available in SNMP Version 1 or Version 2c. SNMP Versions 1 and 2c transmit data between the SNMP server and SNMP agent in clear text. SNMP Version 3 adds authentication and privacy options to secure protocol operations. In addition, this version controls access to the SNMP agent and MIB objects through the User-based Security Model (USM) and View-based Access Control Model (VACM). The ASA also supports the creation of SNMP groups and users, as well as hosts, which is required to enable transport authentication and encryption for secure SNMP communications.

Security Models

For configuration purposes, the authentication and privacy options are grouped together into security models. Security models apply to users and groups, which are divided into the following three types:

- NoAuthPriv—No Authentication and No Privacy, which means that no security is applied to messages.
- AuthNoPriv—Authentication but No Privacy, which means that messages are authenticated.
- AuthPriv—Authentication and Privacy, which means that messages are authenticated and encrypted.

SNMP Groups

An SNMP group is an access control policy to which users can be added. Each SNMP group is configured with a security model, and is associated with an SNMP view. A user within an SNMP group must match the security model of the SNMP group. These parameters specify what type of authentication and privacy a user within an SNMP group uses. Each SNMP group name and security model pair must be unique.

SNMP Users

SNMP users have a specified username, a group to which the user belongs, authentication password, encryption password, and authentication and encryption algorithms to use. The authentication algorithm options are MD5 and SHA. The encryption algorithm options are DES, 3DES, and AES (which is available in 128, 192, and 256 versions). When you create a user, you must associate it with an SNMP group. The user then inherits the security model of the group.

SNMP Hosts

An SNMP host is an IP address to which SNMP notifications and traps are sent. To configure SNMP Version 3 hosts, along with the target IP address, you must configure a username, because traps are only sent to a configured user. SNMP target IP addresses and target parameter names must be unique on the ASA. Each SNMP host can have only one username associated with it. To receive SNMP traps, after you have added the **snmp-server host** command, make sure that you configure the user credentials on the NMS to match the credentials for the ASA.

Implementation Differences Between the ASA and Cisco IOS Software

The SNMP Version 3 implementation in the ASA differs from the SNMP Version 3 implementation in the Cisco IOS software in the following ways:

- The local-engine and remote-engine IDs are not configurable. The local engine ID is generated when the ASA starts or when a context is created.
- No support exists for view-based access control, which results in unrestricted MIB browsing.
- Support is restricted to the following MIBs: USM, VACM, FRAMEWORK, and TARGET.
- You must create users and groups with the correct security model.
- You must remove users, groups, and hosts in the correct sequence.
- Use of the snmp-server host command creates an ASA rule to allow incoming SNMP traffic.

SNMP Syslog Messaging

SNMP generates detailed syslog messages that are numbered 212*nnn*. Syslog messages indicate the status of SNMP requests, SNMP traps, SNMP channels, and SNMP responses from the ASA or ASASM to a specified host on a specified interface.

For detailed information about syslog messages, see the syslog messages guide.

Note SNMP polling fails if SNMP syslog messages exceed a high rate (approximately 4000 per second).

Application Services and Third-Party Tools

For information about SNMP support, see the following URL:

http://www.cisco.com/en/US/tech/tk648/tk362/tk605/tsd_technology_support_sub-protocol_home.html

For information about using third-party tools to walk SNMP Version 3 MIBs, see the following URL:

http://www.cisco.com/en/US/docs/security/asa/asa83/snmp/snmpv3_tools.html

Guidelines for SNMP

This section includes the guidelines and limitations that you should review before configuring SNMP.

Failover and Clustering Guidelines

- The SNMP client in each ASA shares engine data with its peer. Engine data includes the engineID, engineBoots, and engineTime objects of the SNMP-FRAMEWORK-MIB. Engine data is written as a binary file to flash:/snmp/*contextname*.
- When using SNMPv3 with clustering or failover, if you add a new cluster unit after the initial cluster formation or you replace a failover unit, then SNMPv3 users are not replicated to the new unit. You must re-add the SNMPv3 users to the control/active unit to force the users to replicate to the new unit; or you can add the users directly on the new unit (SNMPv3 users and groups are an exception to the rule that you cannot enter configuration commands on a cluster data unit). Reconfigure each user by entering the **snmp-server user** *username group-name* **v3** command on the control/active unit or directly to the data/standby unit with the *priv-password* option and *auth-password* option in their unencrypted forms.

Additional Guidelines

- Power supply traps are not issued for systems operating in appliance mode.
- You must have Cisco Works for Windows or another SNMP MIB-II compliant browser to receive SNMP traps or browse a MIB.
- For secure SNMP polling over a site-to-site VPN, include the IP address of the outside interface in the crypto map access-list as part of the VPN configuration.
- Does not support view-based access control, but the VACM MIB is available for browsing to determine default view settings.

- The ENTITY-MIB is not available in the non-admin context. Use the IF-MIB instead to perform queries in the non-admin context.
- The ENTITY-MIB is not available for the Firepower 9300. Instead, use CISCO-FIREPOWER-EQUIPMENT-MIB and CISCO-FIREPOWER-SM-MIB.
- On some devices, the order of interfaces (ifDescr) in the output of **snmpwalk** has been observed to change after a reboot. The ASA uses an algorithm to determine the ifIndex table that SNMP queries. When the ASA is booted up, the interfaces are added to the ifIndex table in the order loaded as the ASA reads the configuration. New interfaces added to the ASA are appended to the list of interfaces in the ifIndex table. As interfaces are added, removed, or renamed, it can affect the order of interfaces on reboot.
- Does not support SNMP Version 3 for the AIP SSM or AIP SSC.
- Does not support SNMP debugging.
- Does not support retrieval of ARP information.
- Does not support SNMP SET commands.
- When using NET-SNMP Version 5.4.2.1, only supports the encryption algorithm version of AES128. Does not support the encryption algorithm versions of AES256 or AES192.
- Changes to the existing configuration are rejected if the result places the SNMP feature in an inconsistent state.
- For SNMP Version 3, configuration must occur in the following order: group, user, host.
- Before a group is deleted, you must ensure that all users associated with that group are deleted.
- Before a user is deleted, you must ensure that no hosts are configured that are associated with that username.
- If users have been configured to belong to a particular group with a certain security model, and if the security level of that group is changed, you must do the following in this sequence:
    - Remove the users from that group.
    - Change the group security level.
    - Add users that belong to the new group.
- The creation of custom views to restrict user access to a subset of MIB objects is not supported.
- All requests and traps are available in the default Read/Notify View only.
- The connection-limit-reached trap is generated in the admin context. To generate this trap. you must have at least one SNMP server host configured in the user context in which the connection limit has been reached.
- You cannot query for the chassis temperature on the ASA 5585 SSP-40 (NPE).
- You can add up to 4000 hosts. However, only 128 of this number can be for traps.
- The total number of supported active polling destinations is 128.
- You can specify a network object to indicate the individual hosts that you want to add as a host group.
- You can associate more than one user with one host.

- You can specify overlapping network objects in different **host-group** commands. The values that you specify for the last host group take effect for the common set of hosts in the different network objects.
- If you delete a host group or hosts that overlap with other host groups, the hosts are set up again using the values that have been specified in the configured host groups.
- The values that the hosts acquire depend on the specified sequence that you use to run the commands.
- The limit on the message size that SNMP sends is 1472 bytes.
- The ASA supports an unlimited number of SNMP server trap hosts per context. The **show snmp-server host** command output displays only the active hosts that are polling the ASA, as well as the statically configured hosts.

Troubleshooting Tips

- To ensure that the SNMP process that receives incoming packets from the NMS is running, enter the following command:

  ```
  ciscoasa(config)# show process | grep snmp
  ```

- To capture syslog messages from SNMP and have them appear on the ASA console, enter the following commands:

  ```
  ciscoasa(config)# logging list snmp message 212001-212015
  ciscoasa(config)# logging console snmp
  ```

- To make sure that the SNMP process is sending and receiving packets, enter the following commands:

  ```
  ciscoasa(config)# clear snmp-server statistics
  ciscoasa(config)# show snmp-server statistics
  ```

  The output is based on the SNMP group of the SNMPv2-MIB.

- To make sure that SNMP packets are going through the ASA and to the SNMP process, enter the following commands:

  ```
  ciscoasa(config)# clear asp drop
  ciscoasa(config)# show asp drop
  ```

- If the NMS cannot request objects successfully or is not handing incoming traps from the ASA correctly, use a packet capture to isolate the problem, by entering the following commands:

  ```
  ciscoasa (config)# access-list snmp permit udp any eq snmptrap any
  ciscoasa (config)# access-list snmp permit udp any any eq snmp
  ciscoasa (config)# capture snmp type raw-data access-list snmp interface mgmt
  ciscoasa (config)# copy /pcap capture:snmp tftp://192.0.2.5/exampledir/snmp.pcap
  ```

- If the ASA is not performing as expected, obtain information about network topology and traffic by doing the following:
  - For the NMS configuration, obtain the following information:

Number of timeouts

Retry count

Engine ID caching

Username and password used

- Issue the following commands:

  show block

  show interface

  show process

  show cpu

  show vm

- If a fatal error occurs, to help in reproducing the error, send a traceback file and the output of the **show tech-support** command to Cisco TAC.
- If SNMP traffic is not being allowed through the ASA interfaces, you might also need to permit ICMP traffic from the remote SNMP server using the **icmp permit** command.
- For additional troubleshooting information, see the following URL: http://www.cisco.com/c/en/us/support/docs/security/asa-5500-x-series-next-generation-firewalls/116423-troubleshoot-asa-snmp.html

Configure SNMP

This section describes how to configure SNMP.

Procedure

Step 1 Enable the SNMP Agent and SNMP server.

Step 2 Configure SNMP traps.

Step 3 Configure SNMP Version 1 and 2c parameters or SNMP Version 3 parameters.

Enable the SNMP Agent and SNMP Server

To enable the SNMP agent and SNMP server, perform the following steps:

Procedure

Enable the SNMP agent and SNMP server on the ASA. By default, the SNMP server is enabled.

snmp-server enable

Example:

```
ciscoasa(config)# snmp-server enable
```

Configure SNMP Traps

To designate which traps that the SNMP agent generates and how they are collected and sent to NMSs, perform the following steps:

Procedure

Send individual traps, sets of traps, or all traps to the NMS.

snmp-server enable traps [**all** | **syslog** | **snmp** [**authentication** | **linkup** | **linkdown** | **coldstart** | **warmstart**] | **config** | **entity** [**config-change** | **fru-insert** | **fru-remove** | **fan-failure** | **cpu-temperature** | **chassis-fan-failure** | **power-supply**] | **chassis-temperature** | **power-supply-presence** | **power-supply-temperature** | **accelerator-temperature** | **l1-bypass-status**] | **ikev2** [**start** | **stop**] | **ipsec** [**start** | **stop**] | **remote-access** [**session-threshold-exceeded**] | **connection-limit-reached** | **cpu threshold rising** | **interface-threshold** | **memory-threshold** | **nat** [**packet-discard**]

Example:

```
ciscoasa(config)# snmp-server enable traps snmp authentication
linkup linkdown coldstart warmstart
```

This command enables syslog messages to be sent as traps to the NMS. The default configuration has all SNMP standard traps enabled, as shown in the example. To disable these traps, use the **no snmp-server enable traps snmp** command.

If you enter this command and do not specify a trap type, the default is the **syslog** trap. By default, the **syslog** trap is enabled. The default SNMP traps continue to be enabled with the **syslog** trap.

You need to configure both the **logging history** command and the **snmp-server enable traps syslog** command to generate traps from the syslog MIB.

To restore the default enabling of SNMP traps, use the **clear configure snmp-server** command. All other traps are disabled by default.

Traps available in the admin context only:

- **connection-limit-reached**
- **entity**
- **memory-threshold**

Traps generated through the admin context only for physically connected interfaces in the system context:

- **interface-threshold**

All other traps are available in the admin and user contexts in single mode.

In multiple context mode, the **fan-failure** trap, the **power-supply** trap, and the **cpu-temperature** trap are generated only from the admin context, and not the user contexts (applies only to the ASA 5512-X, 5515-X, 5525-X, 5545-X, and 5555-X).

The **config** trap enables the ciscoConfigManEvent notification and the ccmCLIRunningConfigChanged notification, which are generated after you have exited configuration mode.

If the CPU usage is greater than the configured threshold value for the configured monitoring period, the **cpu threshold rising** trap is generated.

When the used system context memory reaches 80 percent of the total system memory, the **memory-threshold** trap is generated from the admin context. For all other user contexts, this trap is generated when the used memory reaches 80 percent of the total system memory in that particular context.

Some traps are not applicable to certain hardware models. Use ? in place of a trap keyword to determine which traps are available for your device. For example:

- The **interface-threshold** trap is not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.
- The **accelerator-temperature** threshold trap applies only to the ASA 5506-X and ASA 5508-X.
- The **chassis-fan-failure** trap does not apply to the ASA 5506-X.
- The following traps do not apply to the ASA 5506-X and ASA 5508-X: **fan-failure**, **fru-insert**, **fru-remove**, **power-supply**, **power-supply-presence**, and **power-supply-temperature**.

Note SNMP does not monitor voltage sensors.

Configure a CPU Usage Threshold

To configure a CPU usage threshold, perform the following steps:

Procedure

Configure the threshold value for a high CPU threshold and the threshold monitoring period.

snmp cpu threshold rising *threshold_value monitoring_period*

Example:

```
ciscoasa(config)# snmp cpu threshold rising 75% 30 minutes
```

To clear the threshold value and monitoring period of the CPU utilization, use the **no** form of this command. If the **snmp cpu threshold rising** command is not configured, the default for the high threshold level is over 70 percent, and the default for the critical threshold level is over 95 percent. The default monitoring period is set to 1 minute.

You cannot configure the critical CPU threshold level, which is maintained at a constant 95 percent. Valid threshold values for a high CPU threshold range from 10 to 94 percent. Valid values for the monitoring period range from 1 to 60 minutes.

Configure a Physical Interface Threshold

To configure the physical interface threshold, perform the following steps:

Procedure

Configure the threshold value for an SNMP physical interface.

snmp interface threshold *threshold_value*

Example:

```
ciscoasa(config)# snmp interface threshold 75%
```

To clear the threshold value for an SNMP physical interface, use the **no** form of this command. The threshold value is defined as a percentage of interface bandwidth utilization. Valid threshold values range from 30 to 99 percent. The default value is 70 percent.

The **snmp interface threshold** command is available only in the admin context.

Physical interface usage is monitored in single mode and multimode, and traps for physical interfaces in the system context are sent through the admin context. Only physical interfaces are used to compute threshold usage.

Note This command is not supported on the ASA Services Module for Catalyst 6500 switches/7600 routers.

Configure Parameters for SNMP Version 1 or 2c

To configure parameters for SNMP Version 1 or 2c, perform the following steps:

Procedure

Step 1 Specify the recipient of an SNMP notification, indicate the interface from which traps are sent, and identify the name and IP address of the NMS or SNMP manager that can connect to the ASA.

snmp-server host{*interface hostname* | *ip_address*} [**trap**| **poll**] [**community** *community-string*] [**version** {**1 2c**| *username*}] [**udp-port** *port*]

Example:

The **trap** keyword limits the NMS to receiving traps only. The **poll** keyword limits the NMS to sending requests (polling) only. By default, SNMP traps are enabled. By default, the UDP port is 162. The community string is a shared secret key between the ASA and the NMS. The key is a case-sensitive value up to 32 alphanumeric characters long. Spaces are not permitted. The default community string is public. The ASA uses this key to determine whether or not the incoming SNMP request is valid. For example, you could designate a site with a community string and then configure the ASA and the management station with the same string. The ASA uses the specified string and do not respond to requests with an invalid community string. After you have used an encrypted community string, only the encrypted form is visible to all systems (for example, CLI, ASDM, CSM, and so on). The clear text password is not visible. The encrypted community string is always generated by the ASA; you normally enter the clear text form.

To receive traps after you have added the **snmp-server host** command, make sure that you configure the user on the NMS with the same credentials as the credentials configured on the ASA.

Step 2 Set the community string, which is for use *only* with SNMP Version 1 or 2c.

snmp-server community *community-string*

Example:

```
ciscoasa(config)# snmp-server community onceuponatime
```

Note You should avoid the use of special characters (!, @, #, $, %, ^, &, *, \) in community strings. In general, using any special characters reserved for functions used by the operating system can cause unexpected results. For example, the backslash (\) is interpreted as an escape character and should not be used in the community string.

Step 3 Set the SNMP server location or contact information.

snmp-server [**contact** | **location**] *text*

Example:

```
ciscoasa(config)# snmp-server location building 42
ciscoasa(config)# snmp-server contact EmployeeA
```

The *text* argument specifies the name of the contact person or the ASA system administrator. The name is case sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.

Step 4 Set the listening port for SNMP requests.

snmp-server listen-port *lport*

Example:

```
ciscoasa(config)# snmp-server lport 192
```

The *lport* argument is the port on which incoming requests are accepted. The default listening port is 161. The **snmp-server listen-port** command is only available in admin context, and is not available in the system context. If you configure the **snmp-server listen-port** command on a port that is currently in use, the following message appears:

```
The UDP port port is in use by another feature. SNMP requests to the device
will fail until the snmp-server listen-port command is configured to use a different port.
```

The existing SNMP thread continues to poll every 60 seconds until the port is available, and issues syslog message %ASA-1-212001 if the port is still in use.

Configure Parameters for SNMP Version 3

To configure parameters for SNMP Version 3, perform the following steps:

Procedure

Step 1 Specify a new SNMP group, which is for use *only* with SNMP Version 3.

snmp-server group *group-name* **v3** [**auth** | **noauth** | **priv**]

Example:

```
ciscoasa(config)# snmp-server group testgroup1 v3 auth
```

When a community string is configured, two additional groups with the name that matches the community string are autogenerated: one for the Version 1 security model and one for the Version 2 security model. The **auth** keyword enables packet authentication. The **noauth** keyword indicates no packet authentication or encryption is being used. The **priv** keyword enables packet encryption and authentication. No default values exist for the **auth** or **priv** keywords.

Step 2 Configure a new user for an SNMP group, which is for use only with SNMP Version 3.

snmp-server user *username group_name* **v3** [**engineID** *engineID*] [**encrypted**] [**auth** {**md5** | **sha**} *auth_password* [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv_password*]]

Example:

```
ciscoasa(config)# snmp-server user testuser1 testgroup1 v3 auth md5 testpassword
aes 128 mypassword
ciscoasa(config)# snmp-server user testuser1 public v3 encrypted auth md5
00:11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF
```

The username argument is the name of the user on the host that belongs to the SNMP agent. The group-name argument is the name of the group to which the user belongs. The **v3** keyword specifies that the SNMP Version 3 security model should be used and enables the use of the **encrypted**, **priv**, and the **auth** keywords. The **engineID** keyword is optional and specifies the engineID of the ASA which was used to localize the user's authentication and encryption information. The engineID argument must specify a valid ASA engineID.

The **encrypted** keyword specifies the password in encrypted format. Encrypted passwords must be in hexadecimal format.

The **auth** keyword specifies which authentication level (**md5** or **sha**) should be used. The **priv** keyword specifies the encryption level. No default values for the **auth** or **priv** keywords, or default passwords exist.

For the encryption algorithm, you can specify the **des**, **3des**, or **aes** keyword. You can also specify which version of the AES encryption algorithm to use: **128**, **192**, or **256**. The auth-password argument specifies the authentication user password. The priv-password argument specifies the encryption user password.

If you forget a password, you cannot recover it and you must reconfigure the user. You can specify a plain-text password or a localized digest. The localized digest must match the authentication algorithm selected for the user, which can be MD5 or SHA. When the user configuration is displayed on the console or is written to a file (for example, the startup-configuration file), the localized authentication and privacy digests are always displayed instead of a plain-text password (see the second example). The minimum length for a password is 1 alphanumeric character; however, we recommend that you use at least 8 alphanumeric characters for security.

When using SNMPv3 with clustering, if you add a new cluster unit after the initial cluster formation, then SNMPv3 users are not replicated to the new unit. You must re-add the SNMPv3 users to the control unit to force the users to replicate to the new unit; or you can add the users directly on the new unit (SNMPv3 users and groups are an exception to the rule that you cannot enter configuration commands on a cluster data unit). Reconfigure each user by entering the **snmp-server user** *username group-name* **v3** command on the control unit or directly to the data unit with the *priv-password* option and *auth-password* option in their unencrypted forms.

If you enter a user on the control unit with the **encrypted** keyword, an error message appears to inform you that the SNMPv3 user commands will not be replicated. This behavior also means that existing SNMPv3 user and group commands are not cleared during replication.

For example, a control unit using commands entered with encrypted keys:

```
ciscoasa(config)# snmp-server user defe abc v3 encrypted auth sha
c0:e7:08:50:47:eb:2e:e4:3f:a3:bc:45:f6:dd:c3:46:25:a0:22:9a
priv aes 256 cf:ad:85:5b:e9:14:26:ae:8f:92:51:12:91:16:a3:ed:de:91:6b:f7:
f6:86:cf:18:c0:f0:47:d6:94:e5:da:01
ERROR: This command cannot be replicated because it contains localized keys.
```

For example, a data unit during cluster replication (appears only if an **snmp-server user** commands exist in the configuration):

```
ciscoasa(cfg-cluster)#
Detected Cluster Master.
Beginning configuration replication from Master.
WARNING: existing snmp-server user CLI will not be cleared.
```

Step 3 Specify the recipient of an SNMP notification. Indicate the interface from which traps are sent. Identify the name and IP address of the NMS or SNMP manager that can connect to the ASA.

snmp-server host *interface* {*hostname* | *ip_address*} [**trap**| **poll**] [**community** *community-string*] [**version** {**1** | **2c** | **3** *username*}] [**udp-port** *port*]

Example:

The **trap** keyword limits the NMS to receiving traps only. The **poll** keyword limits the NMS to sending requests (polling) only. By default, SNMP traps are enabled. By default, the UDP port is 162. The community string is a shared secret key between the ASA and the NMS. The key is a case-sensitive value up to 32 alphanumeric characters. Spaces are not permitted. The default community-string is public. The ASA uses this key to determine whether the incoming SNMP request is valid. For example, you could designate a site with a community string and then configure the ASA and the NMS with the same string. The ASA uses the specified string and do not respond to requests with an invalid community string. After you have used an encrypted community string, only the encrypted form is visible to all systems (for example, CLI, ASDM, CSM, and so on). The clear text password is not visible. The encrypted community string is always generated by the ASA; you normally enter the clear text form.

The **version** keyword specifies the SNMP trap version. The ASA does not support filtering based on SNMP requests (polling).

When SNMP Version 3 hosts are configured on the ASA, a user must be associated with that host.

To receive traps after you have added the **snmp-server host** command, make sure that you configure the user on the NMS with the same credentials as the credentials configured on the ASA.

Step 4 Set the SNMP server location or contact information.

snmp-server [**contact** | **location**] *text*

Example:

```
ciscoasa(config)# snmp-server location building 42
ciscoasa(config)# snmp-server contact EmployeeA
```

The *text* argument specifies the name of the contact person or the ASA system administrator. The name is case sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.

Step 5 Set the listening port for SNMP requests.

snmp-server listen-port *lport*

Example:

```
ciscoasa(config)# snmp-server lport 192
```

The *lport* argument is the port on which incoming requests are accepted. The default listening port is 161. The **snmp-server listen-port** command is only available in admin context, and is not available in the system context. If you configure the **snmp-server listen-port** command on a port that is currently in use, the following message appears:

```
The UDP port port is in use by another feature. SNMP requests to the device
will fail until the snmp-server listen-port command is configured to use a different port.
```

The existing SNMP thread continues to poll every 60 seconds until the port is available, and issues syslog message %ASA-1-212001 if the port is still in use.

Configure a Group of Users

To configure an SNMP user list with a group of specified users in it, perform the following steps:

Procedure

Configure an SNMP user list.

snmp-server user-list *list_name* **username** *user_name*

Example:

```
ciscoasa(config)# snmp-server user-list engineering username user1
```

The *listname* argument specifies the name of the user list, which may be up to 33 characters long. The **username** *user_name* keyword-argument pair specifies the users who may be configured in the user list. You configure the users in the user list with the **snmp-server user** *username* command, which is available only if you are using SNMP Version 3. The user list must have more than one user in it and can be associated with a hostname or a range of IP addresses.

Associate Users with a Network Object

To associate a single user or a group of users in a user list with a network object, perform the following steps:

Procedure

Associate a single user or a group of users in a user list with a network object.

snmp-server host-group *net_obj_name* [**trap**| **poll**] [**community** *community-string*] [**version** {**1** | **2c** | **3** {*username* | **user-list** *list_name*}] [**udp-port** *port*]

Example:

```
ciscoasa(config)# snmp-server host-group inside net1 trap community public version 1
ciscoasa(config)# snmp-server host-group inside net1 trap community public version 2c
ciscoasa(config)# snmp-server host-group inside net1 trap version 3 user1
ciscoasa(config)# snmp-server host-group inside net1 trap version 3 user-list engineering
```

The *net_obj_name* argument specifies the interface network object name with which a user or group of users is associated.

The **trap** keyword specifies that only traps can be sent, and that this host is not allowed to browse (poll). SNMP traps are enabled by default.

The **poll** keyword specifies that the host is allowed to browse (poll), but no traps can be sent.

The **community** keyword specifies that a non-default string is required for requests from the NMS, or when generating traps sent to the NMS. You can use this keyword only for SNMP Version 1 or 2c. The *community-string* argument specifies the password-like community string that is sent with the notification or in a request from the NMS. The community string can have a maximum of 32 characters.

The **version** keyword sets the SNMP notification version to Version 1, 2c, or 3 to use for sending traps and accepting requests (polling). The default version is 1.

The *username* argument specifies the name of the user if you are using SNMP Version 3.

The **user-list** *list_name* keyword-argument pair specifies the name of the user list.

The **udp-port** *port* keyword-argument pair specifies that SNMP traps must be sent to an NMS host on a non-default port and sets the UDP port number of the NMS host. The default UDP port is 162.

Monitoring SNMP

See the following commands for monitoring SNMP.

- **show running-config snmp-server** [**default**]

  This command shows all SNMP server configuration information.

- **show running-config snmp-server group**

  This command shows SNMP group configuration settings.

- **show running-config snmp-server host**

  This command shows configuration settings used by SNMP to control messages and notifications sent to remote hosts.

- **show running-config snmp-server host-group**

  This command shows SNMP host group configurations.

- **show running-config snmp-server user**

  This command shows SNMP user-based configuration settings.

- **show running-config snmp-server user-list**

  This command shows SNMP user list configurations.

- **show snmp-server engineid**

  This command shows the ID of the SNMP engine configured.

- **show snmp-server group**

  This command shows the names of configured SNMP groups. If the community string has already been configured, two extra groups appear by default in the output. This behavior is normal.

- **show snmp-server statistics**

  This command shows the configured characteristics of the SNMP server. To reset all SNMP counters to zero, use the **clear snmp-server statistics** command.

- **show snmp-server user**

  This command shows the configured characteristics of users.

Examples

The following example shows how to display SNMP server statistics:

```
ciscoasa(config)# show snmp-server statistics
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Get-bulk PDUs
    0 Set-request PDUs (Not supported)
0 SNMP packets output
    0 Too big errors (Maximum packet size 512)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs
```

The following example shows how to display the SNMP server running configuration:

```
ciscoasa(config)# show running-config snmp-server
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
```

Examples for SNMP

The following section provides examples that you can use as reference for all SNMP versions.

SNMP Versions 1 and 2c

The following example shows how the ASA can receive SNMP requests from host 192.0.2.5 on the inside interface but does not send any SNMP syslog requests to any host:

```
ciscoasa(config)# snmp-server host 192.0.2.5
ciscoasa(config)# snmp-server location building 42
ciscoasa(config)# snmp-server contact EmployeeA
ciscoasa(config)# snmp-server community ohwhatakeyisthee
```

SNMP Version 3

The following example shows how the ASA can receive SNMP requests using the SNMP Version 3 security model, which requires that the configuration follow this specific order: group, followed by user, followed by host:

```
ciscoasa(config)# snmp-server group v3 vpn-group priv
ciscoasa(config)# snmp-server user admin vpn group v3 auth sha letmein priv 3des cisco123
ciscoasa(config)# snmp-server host mgmt 10.0.0.1 version 3 priv admin
```

History for SNMP

Table 56: History for SNMP

Feature Name	Version	Description
SNMP Versions 1 and 2c	7.0(1)	Provides ASA network monitoring and event information by transmitting data between the SNMP server and SNMP agent through the clear text community string.
SNMP Version 3	8.2(1)	Provides 3DES or AES encryption and support for SNMP Version 3, the most secure form of the supported security models. This version allows you to configure users, groups, and hosts, as well as authentication characteristics by using the USM. In addition, this version allows access control to the agent and MIB objects and includes additional MIB support. We introduced or modified the following commands: **show snmp-server engineid**, **show snmp-server group**, **show snmp-server user**, **snmp-server group**, **snmp-server user**, **snmp-server host**.
Password encryption	8.3(1)	Supports password encryption. We modified the following commands: **snmp-server community**, **snmp-server host**.
SNMP traps and MIBs	8.4(1)	Supports the following additional keywords: **connection-limit-reached**, **cpu threshold rising**, **entity cpu-temperature**, **entity fan-failure**, **entity power-supply**, **ikev2 stop | start**, **interface-threshold**, **memory-threshold**, **nat packet-discard**, **warmstart**. The entPhysicalTable reports entries for sensors, fans, power supplies, and related components. Supports the following additional MIBs: CISCO-ENTITY-SENSOR-EXT-MIB, CISCO-ENTITY-FRU-CONTROL-MIB, CISCO-PROCESS-MIB, CISCO-ENHANCED-MEMPOOL-MIB, CISCO-L4L7MODULE-RESOURCE-LIMIT-MIB, DISMAN-EVENT-MIB, DISMAN-EXPRESSION-MIB, ENTITY-SENSOR-MIB, NAT-MIB. Supports the following additional traps: ceSensorExtThresholdNotification, clrResourceLimitReached, cpmCPURisingThreshold, mteTriggerFired, natPacketDiscard, warmStart. We introduced or modified the following commands: **snmp cpu threshold rising**, **snmp interface threshold**, **snmp-server enable traps**.
IF-MIB ifAlias OID support	8.2(5) / 8.4(2)	The ASA now supports the ifAlias OID. When you browse the IF-MIB, the ifAlias OID will be set to the value that has been set for the interface description.

Feature Name	Version	Description
ASA Services Module (ASASM)	8.5(1)	The ASASM supports all MIBs and traps that are present in 8.4(1), except for the following: Unsupported MIBs in 8.5(1): • CISCO-ENTITY-SENSOR-EXT-MIB (Only objects under the entPhySensorTable group are supported). • ENTITY-SENSOR-MIB (Only objects in the entPhySensorTable group are supported). • DISMAN-EXPRESSION-MIB (Only objects in the expExpressionTable, expObjectTable, and expValueTable groups are supported). Unsupported traps in 8.5(1): • ceSensorExtThresholdNotification (CISCO-ENTITY-SENSOR-EXT-MIB). This trap is only used for power supply failure, fan failure, and high CPU temperature events. • InterfacesBandwidthUtilization.
SNMP traps	8.6(1)	Supports the following additional keywords for the ASA 5512-X, 5515-X, 5525-X, 5545-X, and 5555-X: **entity power-supply-presence**, **entity power-supply-failure**, **entity chassis-temperature**, **entity chassis-fan-failure, entity power-supply-temperature**. We modified the following command: **snmp-server enable traps**.
VPN-related MIBs	9.0(1)	An updated version of the CISCO-IPSEC-FLOW-MONITOR-MIB.my MIB has been implemented to support the next generation encryption feature. The following MIBs have been enabled for the ASASM: • ALTIGA-GLOBAL-REG.my • ALTIGA-LBSSF-STATS-MIB.my • ALTIGA-MIB.my • ALTIGA-SSL-STATS-MIB.my • CISCO-IPSEC-FLOW-MONITOR-MIB.my • CISCO-REMOTE-ACCESS-MONITOR-MIB.my
Cisco TrustSec MIB	9.0(1)	Support for the following MIB was added: CISCO-TRUSTSEC-SXP-MIB.
SNMP OIDs	9.1(1)	Five new SNMP Physical Vendor Type OIDs have been added to support the ASA 5512-X, 5515-X, 5525-X, 5545-X, and 5555-X.
NAT MIB	9.1(2)	Added the cnatAddrBindNumberOfEntries and cnatAddrBindSessionCount OIDs to support the xlate_count and max_xlate_count entries, which are the equivalent to allowing polling using the **show xlate count** command.

Feature Name	Version	Description
SNMP hosts, host groups, and user lists	9.1(5)	You can now add up to 4000 hosts. The number of supported active polling destinations is 128. You can specify a network object to indicate the individual hosts that you want to add as a host group. You can associate more than one user with one host. We introduced or modified the following commands: **snmp-server host-group**, **snmp-server user-list**, **show running-config snmp-server**, **clear configure snmp-server**.
SNMP message size	9.2(1)	The limit on the message size that SNMP sends has been increased to 1472 bytes.
SNMP OIDs and MIBs	9.2(1)	The ASA now supports the cpmCPUTotal5minRev OID. The ASAv has been added as a new product to the SNMP sysObjectID OID and entPhysicalVendorType OID. The CISCO-PRODUCTS-MIB and CISCO-ENTITY-VENDORTYPE-OID-MIB have been updated to support the new ASAv platform. A new SNMP MIB for monitoring VPN shared license usage has been added.
SNMP OIDs and MIBs	9.3(1)	CISCO-REMOTE-ACCESS-MONITOR-MIB (OID 1.3.6.1.4.1.9.9.392) support has been added for the ASASM.
SNMP MIBs and traps	9.3(2)	The CISCO-PRODUCTS-MIB and CISCO-ENTITY-VENDORTYPE-OID-MIB have been updated to support the ASA 5506-X. The ASA 5506-X has been added as new products to the SNMP sysObjectID OID and entPhysicalVendorType OID tables. The ASA now supports the CISCO-CONFIG-MAN-MIB, which enables you to do the following: • Know which commands have been entered for a specific configuration. • Notify the NMS when a change has occurred in the running configuration. • Track the time stamps associated with the last time that the running configuration was changed or saved. • Track other changes to commands, such as terminal details and command sources. We modified the following command: **snmp-server enable traps**.
SNMP MIBs and traps	9.4(1)	The ASA 5506W-X, ASA 5506H-X, ASA 5508-X, and ASA 5516-X have been added as a new product to the SNMP sysObjectID OID and entPhysicalVendorType OID tables.

Feature Name	Version	Description
Unlimited SNMP server trap hosts per context	9.4(1)	The ASA supports unlimited SNMP server trap hosts per context. The **show snmp-server host** command output displays only the active hosts that are polling the ASA, as well as the statically configured hosts. We modified the following command: **show snmp-server host**.
Added support for ISA 3000	9.4(1.225)	The ISA 3000 family of products is now supported for SNMP. We added new OIDs for this platform. The **snmp-server enable traps entity** command has been modified to include a new variable *l1-bypass-status*. This enables hardware bypass status change. We modified the following command: **snmp-server enable traps entity**.

CHAPTER 42

Anonymous Reporting and Smart Call Home

This chapter describes how to configure the Anonymous Reporting and Smart Call Home services.

About Anonymous Reporting

You can help to improve the Cisco ASA platform by enabling Anonymous Reporting, which allows Cisco to securely receive minimal error and health information from the device. If you enable the feature, your customer identity will remain anonymous, and no identifying information will be sent.

Enabling Anonymous Reporting creates a trust point and installs a certificate. A CA certificate is required for your ASA to validate the server certificate present on the Smart Call Home web server and to form the HTTPS session so that your ASA can send messages securely. Cisco imports a certificate that is predefined in the software. If you decide to enable Anonymous Reporting, a certificate is installed on the ASA with a hardcoded trust point name: _SmartCallHome_ServerCA. When you enable Anonymous Reporting, this trust point is created, the appropriate certificate is installed, and you receive a message about this action. The certificate then appears in your configuration.

If the appropriate certificate already exists in your configuration when you enable Anonymous Reporting, no trust point is created, and no certificate is installed.

Note When you enable Anonymous Reporting, you acknowledge your consent to transfer the specified data to Cisco or to vendors operating on Cisco's behalf (including countries outside of the U.S.). Cisco maintains the privacy of all customers. For information about Cisco's treatment of personal information, see the Cisco Privacy Statement at the following URL: http://www.cisco.com/web/siteassets/legal/privacy.html

DNS Requirement

A DNS server must be configured correctly for the ASA to reach the Cisco Smart Call Home server and send messages to Cisco. Because it is possible that the ASA resides in a private network and does not have access to the public network, Cisco verifies your DNS configuration and then configures it for you, if necessary, by doing the following:

1. Performing a DNS lookup for all DNS servers configured.
2. Getting the DNS server from the DHCP server by sending DHCPINFORM messages on the highest security-level interface.
3. Using the Cisco DNS servers for lookup.
4. Randomly using a static IP addresses for tools.cisco.com.

These tasks are performed without changing the current configuration. (For example, the DNS server that was learned from DHCP will not be added to the configuration.)

If there is no DNS server configured, and the ASA cannot reach the Cisco Smart Call Home Server, Cisco generates a syslog message with the warning severity level for each Smart Call Home message that is sent to remind you to configure DNS correctly.

See the syslog messages guide for information about syslog messages.

About Smart Call Home

When fully configured, Smart Call Home detects issues at your site and reports them back to Cisco or through other user-defined channels (such as e-mail or directly to you), often before you know that these issues exist. Depending on the seriousness of these problems, Cisco responds to your system configuration issues, product end-of-life announcements, security advisory issues, and so on by providing the following services:

- Identifying issues quickly with continuous monitoring, real-time proactive alerts, and detailed diagnostics.
- Making you aware of potential problems through Smart Call Home notifications, in which a service request has been opened, with all diagnostic data attached.
- Resolving critical problems faster with direct, automatic access to experts in Cisco TAC.
- Using staff resources more efficiently by reducing troubleshooting time.
- Generating service requests to Cisco TAC automatically (if you have a service contract), routed to the appropriate support team, which provides detailed diagnostic information that speeds problem resolution.

The Smart Call Home Portal offers quick access to required information that enables you to do the following:

- Review all Smart Call Home messages, diagnostics, and recommendations in one place.
- Check service request status.
- View the most up-to-date inventory and configuration information for all Smart Call Home-enabled devices.

Subscribe to Alert Groups

An alert group is a predefined subset of the Smart Call Home alerts that are supported on the ASA. Different types of Smart Call Home alerts are grouped into different alert groups, depending on their type. Each alert group reports the output of certain CLIs. The supported Smart Call Home alert groups are the following:

- syslog
- diagnostic
- environment
- inventory
- configuration
- threat
- snapshot
- telemetry
- test

Attributes of Alert Groups

Alert groups have the following attributes:

- Events first register with one alert group.
- A group can associate with multiple events.
- You can subscribe to specific alert groups.
- You can enable and disable specific alert groups. The default setting is enabled for all alert groups.
- The diagnostic and environment alert groups support subscription for periodic messages.
- The syslog alert group supports message ID-based subscription.
- You can configure a threshold for CPU and memory usage for the environment alert group. When a certain parameter has exceeded a predefined threshold, a message is sent. Most of the threshold values are platform-dependent and cannot be changed.
- You configure the snapshot alert group to send the output of CLIs that you specify.

Messages Sent to Cisco by Alert Groups

Messages are sent to Cisco periodically and whenever the ASA reloads. These messages are categorized by alert groups.

Inventory alerts consist of output from the following commands:

- **show version**—Displays the ASA software version, hardware configuration, license key, and related uptime data for the device.
- **show inventory—**Retrieves and displays inventory information about each Cisco product that is installed in the networking device. Each product is identified by unique device information, called the UDI, which

is a combination of three separate data elements: the product identifier (PID), the version identifier (VID), and the serial number (SN).

- **show failover state**—Displays the failover state of both units in a failover pair. The information displayed includes the primary or secondary status of the unit, the Active/Standby status of the unit, and the last reported reason for failover.
- **show module**—Shows information about any modules installed on the ASAs, for example, information about an SSP installed on the ASA 5585-X, and information about an IPS SSP installed on an ASA 5585-X.
- **show environment**—Shows system environment information for ASA system components, such as hardware operational status for the chassis, drivers, fans, and power supplies, as well as temperature status, voltage, and CPU usage.

Configuration alerts consist of output from the following commands:

- **show context**—Shows allocated interfaces and the configuration file URL, the number of contexts configured, or if you enable Anonymous Reporting in the system execution space, from a list of all contexts.
- **show call-home registered-module status**—Shows the registered module status. If you use system configuration mode, the command displays system module status based on the entire device, not per context.
- **show running-config**—Shows the configuration that is currently running on the ASA.
- **show startup-config**—Show the startup configuration.
- **show access-list | include elements**—Shows the hit counters and a time stamp value for an access list.

Diagnostic alerts consist of output from the following commands:

- **show failover**—Displays information about the failover status of the unit.
- **show interface—Displays interface statistics.**
- **show cluster info—Displays cluster information.**
- **show cluster history—Displays the cluster history.**
- **show crashinfo** (truncated)—After an unexpected software reload, the device sends a modified crash information file with only the traceback section of the file included, so only function calls, register values, and stack dumps are reported to Cisco.
- **show tech-support no-config—Displays the information that is used for diagnosis by technical support analysts.**

Environment alerts consist of output from the following commands:

- **show environment**—Shows system environment information for ASA system components, such as hardware operational status for the chassis, drivers, fans, and power supplies, as well as temperature status, voltage, and CPU usage.
- **show cpu usage**—Displays CPU usage information.
- **show memory detail**—Displays details of the free and allocated system memory.

Threat alerts consist of output from the following commands:

- **show threat-detection rate**—Displays threat detection statistics.
- **show threat-detection shun**—Displays currently shunned hosts.
- **show shun**—Displays shun information.
- **show dynamic-filter reports top**—Generates reports of the top 10 malware sites, ports, and infected hosts classified by the Botnet Traffic Filter.

Snapshot alerts may consist of output from the following commands:

- **show conn count**—Shows the number of active connections.
- **show asp drop**—Shows the accelerated security path dropped packets or connections.

Telemetry alerts consist of output from the following commands:

- **show perfmon detail—Shows** ASA performance details.
- **show traffic—Displays interface transmit and receive activity.**
- **show conn count**—Shows the number of active connections.
- **show vpn-sessiondb summary—Shows VPN session summary information.**
- **show vpn load-balancing—Displays the runtime statistics for the VPN load-balancing virtual cluster configuration.**
- **show local-host | include interface—Shows the network states of local hosts.**
- **show memory—Displays a summary of the maximum physical memory and current free memory available to the operating system.**
- **show context**—Shows allocated interfaces and the configuration file URL, the number of contexts configured, or if you enable Anonymous Reporting in the system execution space, from a list of all contexts.
- **show access-list | include elements**—Shows the hit counters and a time stamp value for an access list.
- **show interface—Displays interface statistics.**
- **show threat-detection statistics protocol—Shows IP protocol statistics.**
- **show phone-proxy media-sessions count—Displays the number of corresponding media sessions stored by the Phone Proxy.**
- **show phone-proxy secure-phones count—Displays the number of phones capable of secure mode stored in the database.**
- **show route—Displays the routing table.**
- **show xlate count—Shows the number of NAT sessions (xlates).**

Message Severity Threshold

When you subscribe a destination profile to certain alert groups, you can set a threshold for sending alert group messages based on the message severity level. Any message with a value lower than the destination profile's specified threshold is not sent to the destination.

The following table shows the mapping between message severity levels and syslog severity levels.

Table 57: Message Severity Level and Syslog Level Mapping

Level	Message Severity Level	Syslog Severity Level	Description
9	Catastrophic	N/A	Network-wide catastrophic failure.
8	Disaster	N/A	Significant network impact.
7	Determined by the specified CLI keyword: **subscribe-to-alert-group** *name of alert group* **severity** *severity level*	0	Emergency. System is unusable.
6	Determined by the specified CLI keyword: **subscribe-to-alert-group** *name of alert group* **severity** *severity level*	1	Alert. Critical conditions; immediate attention needed.
5	Determined by the specified CLI keyword: **subscribe-to-alert-group** *name of alert group* **severity** *severity level*	2	Critical. Major conditions.
4	Determined by the specified CLI keyword: **subscribe-to-alert-group** *name of alert group* **severity** *severity level*	3	Error. Minor conditions.
3	Warning	4	Warning conditions.
2	Notification	5	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	6	Information. Normal event, signifying a return to normal state.
0	Debugging	7	Debugging messages (default setting).

Subscription Profiles

A subscription profile allows you to associate the destination recipients with interested groups. When an event registered with a subscribed group in a profile is triggered, the message associated with the event is sent to the configured recipients. Subscription profiles have the following attributes:

- You can create and configure multiple profiles.
- A profile may configure multiple e-mail or HTTPS recipients.
- A profile may subscribe multiple groups to a specified severity level.
- A profile supports three message formats: short text, long text, and XML.
- You can enable and disable a specific profile. Profiles are disabled by default.
- You can specify the maximum message size. The default is 3 MB.

A default profile, “Cisco TAC,” has been provided. The default profile has a predefined set of groups (diagnostic, environment, inventory, configuration, and telemetry) to monitor and predefined destination e-mail and HTTPS URLs. The default profile is created automatically when you initially configure Smart Call Home. The destination e-mail is callhome@cisco.com and the destination URL is https://tools.cisco.com/its/service/oddce/services/DDCEService.

Note You cannot change the destination e-mail or the destination URL of the default profile.

When you subscribe a destination profile to the configuration, inventory, telemetry, or snapshot alert groups, you can choose to receive the alert group messages asynchronously or periodically at a specified time.

The following table maps the default alert group to its severity level subscription and period (if applicable):

Table 58: Alert Group to Severity Level Subscription Mapping

Alert Group	Severity Level	Period
Configuration	Informational	Monthly
Diagnostic	Informational and higher	N/A
Environment	Notification and higher	N/A
Inventory	Informational	Monthly
Snapshot	Informational	N/A
Syslog	Equivalent syslog	N/A
Telemetry	Informational	Daily
Test	N/A	N/A
Threat	Notification	N/A

Guidelines for Anonymous Reporting and Smart Call Home

This section includes the guidelines and limitation that you should review before configuring Anonymous reporting and Smart Call Home.

Anonymous Reporting Guidelines

- DNS must be configured.
- If an Anonymous Reporting message cannot be sent on the first try, the ASA retries two more times before dropping the message.
- Anonymous Reporting may coexist with other Smart Call Home configurations without changing the existing configuration. For example, if Smart Call Home is disabled before enabling Anonymous Reporting, it remains disabled, even after Anonymous Reporting has been enabled.
- If Anonymous Reporting is enabled, you cannot remove the trust point, and when Anonymous Reporting is disabled, the trust point remains. If Anonymous Reporting is disabled, you can remove the trust point, but disabling Anonymous Reporting does not cause the trust point to be removed.
- If you are using a multiple context mode configuration, the **dns**, **interface**, and **trustpoint** commands are in the admin context, and the **call-home** commands are in the system context.
-

Smart Call Home Guidelines

- In multiple context mode, the subscribe-to-alert-group snapshot periodic command is divided into two commands: one to obtain information from the system configuration and one to obtain information from the user context.
- The Smart Call Home back-end server can accept messages in XML format only.
- A Smart Call Home message is sent to Cisco to report important cluster events if you have enabled clustering and configured Smart Call Home to subscribe to the diagnostic alert group with a critical severity level. A Smart Call Home clustering message is sent for only the following events:
  - When a unit joins the cluster
  - When a unit leaves the cluster
  - When a cluster unit becomes the cluster control unit
  - When a secondary unit fails in the cluster

  Each message that is sent includes the following information:

  - The active cluster member count
  - The output of the **show cluster info** command and the **show cluster history** command on the cluster control unit

Configure Anonymous Reporting and Smart Call Home

While Anonymous Reporting is part of the Smart Call Home service and allows Cisco to anonymously receive minimal error and health information from your device, the Smart Call Home service provides customized support of your system health, enabling Cisco TAC to monitor your devices and open a case when there is an issue, often before you know the issue has occurred.

You can have both services configured on your system at the same time, although configuring the Smart Call Home service provides the same functionality as Anonymous Reporting, plus customized services.

When you enter configuration mode, you receive a prompt that requests you to enable the Anonymous Reporting and Smart Call Home services according to the following guidelines:

- At the prompt, you may choose [Y]es, [N]o, or [A]sk later. If you choose [A]sk later, then you are reminded again in seven days or when the ASA reloads. If you continue to choose [A]sk later, the ASA prompts two more times at seven-day intervals before it assumes a [N]o response and does not ask again.
- If you did not receive the prompt, you may enable Anonymous Reporting or Smart Call Home by performing the steps in Configure Anonymous Reporting, on page 1037 or in Configure Smart Call Home, on page 1038.

Configure Anonymous Reporting

To configure Anonymous Reporting, perform the following steps:

Procedure

Step 1 Enable the Anonymous Reporting feature and create a new anonymous profile.

call-home reporting anonymous

Example:

```
ciscoasa(config)# call-home reporting anonymous
```

Entering this command creates a trust point and installs a certificate that is used to verify the identity of the Cisco web server.

Step 2 (Optional) Make sure that you have connectivity to the server and that your system can send messages.

call-home test reporting anonymous

Example:

```
ciscoasa(config)# call-home test reporting anonymous

INFO: Sending test message to
https://tools.cisco.com/its/service/oddce/services/DDCEService...

INFO: Succeeded
```

A success or error message returns test results.

Configure Smart Call Home

Configuring the Smart Call Home service on your ASA includes the following tasks:

Procedure

Step 1 Enable the Smart Call Home service. See Enable Smart Call Home, on page 1038.

Step 2 Configure the mail server through which Smart Call Home messages are delivered to subscribers. See Configure the Mail Server, on page 1043.

Step 3 Set up contact information for the Smart Call Home messages. See Configure Customer Contact Information, on page 1041.

Step 4 Define alert processing parameters, such as the maximum rate of events that can be handled. See Configure Alert Group Subscription, on page 1040.

Step 5 Set up alert subscription profiles. See Configure a Destination Profile, on page 1045.

Each alert subscription profile identifies the following:

- The subscribers to whom the Smart Call Home messages are sent, such as a Smart Call Home server at Cisco or a list of e-mail recipients.
- Information categories for which you want to receive alerts, such as configuration or inventory information.

Enable Smart Call Home

To enable Smart Call Home and activate your call-home profile, perform the following steps:

Procedure

Step 1 Enable the Smart Call Home service.

service call-home

Example:

```
ciscoasa(config)# service call-home
```

Step 2 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call home
```

Declare and Authenticate a Certificate Authority Trust Point

If Smart Call Home is configured to send messages to a web server through HTTPS, you need to configure the ASA to trust the certificate of the web server or the certificate of the Certificate Authority (CA) that issued the certificate. The Cisco Smart Call Home Production server certificate is issued by Verisign. The Cisco Smart Call Home Staging server certificate is issued by the Digital Signature Trust Company.

Note You should set the trust point for no client-types/no validation-usage to prevent it from being used for VPN validation.

To declare and authenticate the Cisco server security certificate and establish communication with the Cisco HTTPS server for Smart Call Home service, perform the following steps:

Procedure

Step 1 (Multiple Context Mode only) Install the certificate in the admin context.

changeto context *admincontext*

Example:

```
ciscoasa(config)# changeto context contextA
```

Step 2 Configure a trust point and prepare for certificate enrollment.

crypto ca trustpoint *trustpoint-name*

Example:

```
ciscoasa(config)# crypto ca trustpoint cisco
```

Note If you use HTTP as the transport method, you must install a security certificate through a trust point, which is required for HTTPS. Find the specific certificate to install at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/smart_call_home/SCH31_Ch6.html#wp1035380

Step 3 Specify a manual cut-and-paste method of certificate enrollment.

enroll terminal

Example:

```
ciscoasa(ca-trustpoint)# enroll terminal
```

Step 4 Authenticate the named CA. The CA name should match the trust point name specified in the **crypto ca trustpoint** command. At the prompt, paste the security certificate text.

crypto ca authenticate trustpoint

Example:

```
ciscoasa(ca-trustpoint)# crypto ca authenticate cisco
```

Step 5 Specify the end of the security certificate text and confirm acceptance of the entered security certificate.

quit

Example:

```
ciscoasa(ca-trustpoint)# quit

%Do you accept this certificate [yes/no]:

yes
```

Configure the Environment and Snapshot Alert Groups

To configure the environment and snapshot alert groups, perform the following steps:

Procedure

Enter alert-group-configuration mode.

alert-group-config {**environment** | **snapshot**}

Example:

```
ciscoasa(config)# alert-group-config environment
```

Configure Alert Group Subscription

To subscribe a destination profile to an alert group, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Enable the specified Smart Call Home alert group.

alert-group {**all** | **configuration** | **diagnostic** | **environment** | **inventory** | **syslog**}

Example:

```
ciscoasa(cfg-call-home)# alert-group syslog
```

Use the **all** keyword to enable all alert groups. By default, all alert groups are enabled.

Step 3 Enter the profile configuration mode for the specified destination profile.

profile *profile-name*

Example:

```
ciscoasa(cfg-call-home)# profile CiscoTAC-1
```

Step 4 Subscribe to all available alert groups.

subscribe-to-alert-group all

Example:

```
ciscoasa(cfg-call-home-profile)# subscribe-to-alert-group all
```

Step 5 Subscribe this destination profile to the configuration alert group.

subscribe-to-alert-group configuration periodic {**daily** *hh:mm* | **monthly** *date hh:mm* | **weekly** *day hh:mm*}

Example:

```
ciscoasa(cfg-call-home-profile)# subscribe-to-alert-group configuration periodic weekly
Wednesday 23:30
```

The **periodic** keyword configures the configuration alert group for periodic notification. The default period is daily.

The **daily** keyword specifies the time of the day to send, in the *hh:mm* format, with a 24-hour clock (for example, 14:30).

The **weekly** keyword specifies the day of the week and time of day in the *day hh:mm* format, where the day of the week is spelled out (for example, Monday).

The **monthly** keyword specifies the numeric date, from 1 to 31, and the time of day, in the *date hh:mm* format.

Configure Customer Contact Information

To configure customer contact information, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Specify the customer phone number. Spaces are allowed, but you must use quotes around the string if it includes spaces.

phone-number *phone-number-string*

Example:

```
ciscoasa(cfg-call-home)# phone-number 8005551122
```

Step 3 Specify the customer address, which is a free-format string that may be up to 255 characters long. Spaces are allowed, but you must use quotes around the string if it includes spaces.

street-address *street-address*

Example:

```
ciscoasa(cfg-call-home)# street-address “1234 Any Street, Any city, Any state, 12345”
```

Step 4 Specify the customer name, which may be up to 128 characters long. Spaces are allowed, but you must use quotes around the string if it includes spaces.

contact-name *contact-name*

Example:

```
ciscoasa(cfg-call-home)# contact-name contactname1234
```

Step 5 Specify the Cisco customer ID, which may be up to 64 characters long. Spaces are allowed, but you must use quotes around the string if it includes spaces.

customer-id *customer-id-string*

Example:

```
ciscoasa(cfg-call-home)# customer-id customer1234
```

Step 6 Specify the customer site ID, which may be up to 64 characters long. Spaces are allowed, but you must use quotes around the string if it includes spaces.

site-id *site-id-string*

Example:

```
ciscoasa(cfg-call-home)# site-id site1234
```

Step 7 Specify the customer contract identification, which may be up to 128 characters long. Spaces are allowed, but you must use quotes around the string if it includes spaces.

contract-id *contract-id-string*

Example:

```
ciscoasa(cfg-call-home)# contract-id contract1234
```

Example

The following example shows how to configure contact information:

```
ciscoasa(config)# call-home
ciscoasa(cfg-call-home)# contact-email-addr username@example.com
ciscoasa(cfg-call-home)# phone-number 8005551122
ciscoasa(cfg-call-home)# street-address "1234 Any Street, Any city, Any state, 12345"
ciscoasa(cfg-call-home)# contact-name contactname1234
ciscoasa(cfg-call-home)# customer-id customer1234
ciscoasa(cfg-call-home)# site-id site1234
ciscoasa(cfg-call-home)# contract-id contract1234
```

Configure the Mail Server

We recommend that you use HTTPS for message transport because it is the most secure. However, you may configure an e-mail destination for Smart Call Home and then configure the mail server to use the e-mail message transport.

To configure the mail server, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Specify the SMTP mail server.

mail-server*ip-address* **name priority** [**1-100**] [**all**]

Example:

```
ciscoasa(cfg-call-home)# mail-server 10.10.1.1 smtp.example.com priority 1
```

You can specify up to five mail servers, using five separate commands. You must configure at least one mail server for using e-mail transport of Smart Call Home messages.

The lower the number, the higher the priority of the mail server.

The *ip-address* argument can be an IPv4 or IPv6 mail server address.

Example

The following example shows how to configure a primary mail server (named ”smtp.example.com”) and a secondary mail server at IP address 10.10.1.1:

```
ciscoasa(config)# call-home
ciscoasa(cfg-call-home)# mail-server smtp.example.com priority 1
ciscoasa(cfg-call-home)# mail-server 10.10.1.1 priority 2
ciscoasa(cfg-call-home)# exit
ciscoasa(config)#
```

Configure Traffic Rate Limiting

To configure traffic rate limiting, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Specify the number of messages that Smart Call Home can send per minute. The default value is 10 messages per minute.

rate-limit *msg-count*

Example:

```
ciscoasa(cfg-call-home)# rate-limit 5
```

Send Smart Call Home Communications

To send specific Smart Call Home communications, perform the following steps:

Procedure

Choose one of the following options:

- Option 1—Send a test message manually using a profile configuration.

  call-home test [*test-message*] **profile** *profile-name*

  Example:

  ```
  ciscoasa# call-home test [testing123] profile CiscoTAC-1
  ```

- Option 2—Send an alert group message to one destination profile, if specified. If no profile is specified, send messages to all profiles that are subscribed to the inventory, configuration, snapshot, or telemetry alert groups.

  call-home send alert-group inventory { | **configuration** | **snapshot** | **telemetry**} [**profile** *profile-name*]

  Example:

  ```
  ciscoasa# call-home send alert-group inventory
  ```

- Option 3—Send command output to an e-mail address. The specified CLI command can be any command, including commands for all registered modules.

  call-home send*cli command* [**email** *email*]

  Example:

  ```
  ciscoasa# call-home send cli destination email username@example.com
  ```

  If you specify an e-mail address, the command output is sent to that address. If no e-mail address is specified, the output is sent to Cisco TAC. The e-mail is sent in log text format with the service number, if specified, in the subject line.

  The service number is required only if no e-mail address is specified, or if a Cisco TAC e-mail address is specified.

Configure a Destination Profile

To configure a destination profile for e-mail or for HTTP, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Enter the profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.

profile *profile-name*

Example:

```
ciscoasa(cfg-call-home)# profile newprofile
```

You can create a maximum of 10 active profiles. The default profile is to report back to Cisco TAC. If you want to send call home information to a different location (for example, your own server), you can configure a separate profile.

Step 3 Configure the destination, message size, message format, and transport method for the Smart Call Home message receiver. The default message format is XML, and the default enabled transport method is e-mail.

destination address {**email** *address* | **http** *url*} | **message-size-limit** *size* | **preferred-msg-format** {**long-text** | **short-text** | **xml**} **transport-method** {**email** | **http**}}

Example:

```
ciscoasa(cfg-call-home-profile)# destination address email username@example.com
ciscoasa(cfg-call-home-profile)# destination preferred-msg-format long-text
```

The e-mail-address is the e-mail address of the Smart Call Home message receiver, which can be up to 100 characters long. By default, the maximum URL size is 5 MB.

Use the short-text format to send and read a message on a mobile device, and use the long text format to send and read a message on a computer.

If the message receiver is the Smart Call Home back-end server, ensure that the **preferred-msg-format** value is XML because the back-end server can accept messages in XML format only.

Use this command to change the transport method back to e-mail.

Copy a Destination Profile

To create a new destination profile by copying an existing one, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Specify the profile to copy.

profile *profile-name*

Example:

```
ciscoasa(cfg-call-home)# profile newprofile
```

Step 3 Copy the content of an existing profile to a new profile.

copy profile *src-profile-name dest-profile-name*

Example:

```
ciscoasa(cfg-call-home)# copy profile newprofile profile1
```

The existing profile (*src-profile-name*) and the new profile (*dest-profile-name*) may be up to 23 characters long.

Example

The following example shows how to copy an existing profile:

```
ciscoasa(config)# call-home
ciscoasa(cfg-call-home)# profile newprofile
ciscoasa(cfg-call-home-profile)# copy profile newprofile profile1
```

Rename a Destination Profile

To change the name of an existing profile, perform the following steps:

Procedure

Step 1 Enter call-home configuration mode.

call-home

Example:

```
ciscoasa(config)# call-home
```

Step 2 Specify the profile to rename.

profile *profilename*

Example:

```
ciscoasa(cfg-call-home)# profile newprofile
```

Step 3 Change the name of an existing profile.

rename profile *src-profile-name dest-profile-name*

Example:

```
ciscoasa(cfg-call-home)# rename profile newprofile profile1
```

The existing profile (*src-profile-name*) and the new profile (*dest-profile-name*) may be up to 23 characters long.

Example

The following example shows how to rename an existing profile:

```
ciscoasa(config)# call-home
ciscoasa(cfg-call-home)# profile newprofile
ciscoasa(cfg-call-home-profile)# rename profile newprofile profile1
```

Monitoring Anonymous Reporting and Smart Call Home

See the following commands for monitoring Anonymous Reporting and Smart Call Home services.

- **show call-home detail**

  This command shows the current Smart Call Home detail configuration.

- **show call-home mail-server status**

  This command shows the current mail server status.

- **show call-home profile** {profile name | **all**}

  This command shows the configuration of Smart Call Home profiles.

- **show call-home registered-module status** [**all**]

  This command shows the registered module status.

- **show call-home statistics**

  This command shows call-home detail status.

- **show call-home**

  This command shows the current Smart Call Home configuration.

- **show running-config call-home**

  This command shows the current Smart Call Home running configuration.

- **show smart-call-home alert-group**

  This command shows the current status of Smart Call Home alert groups.

- **show running-config all**

  This command shows details about the Anonymous Reporting user profile.

Examples for Smart Call Home

The following example shows how to configure the Smart Call Home service:

```
ciscoasa (config)# service call-home
ciscoasa (config)# call-home
ciscoasa (cfg-call-home)# contact-email-addr customer@example.com
ciscoasa (cfg-call-home)# profile CiscoTAC-1
ciscoasa (cfg-call-home-profile)# destination address http
https://example.cisco.com/its/service/example/services/ExampleService
ciscoasa (cfg-call-home-profile)# destination address email callhome@example.com
ciscoasa (cfg-call-home-profile)# destination transport-method http
ciscoasa (cfg-call-home-profile)# subscribe-to-alert-group inventory periodic daily 23:30
ciscoasa (cfg-call-home-profile)# subscribe-to-alert-group configuration periodic weekly
Wednesday 23:30
ciscoasa (cfg-call-home-profile)# subscribe-to-alert-group environment
ciscoasa (cfg-call-home-profile)# subscribe-to-alert-group diagnostic
ciscoasa (cfg-call-home-profile)# subscribe-to-alert-group telemetry periodic weekly Monday
 23:30
```

History for Anonymous Reporting and Smart Call Home

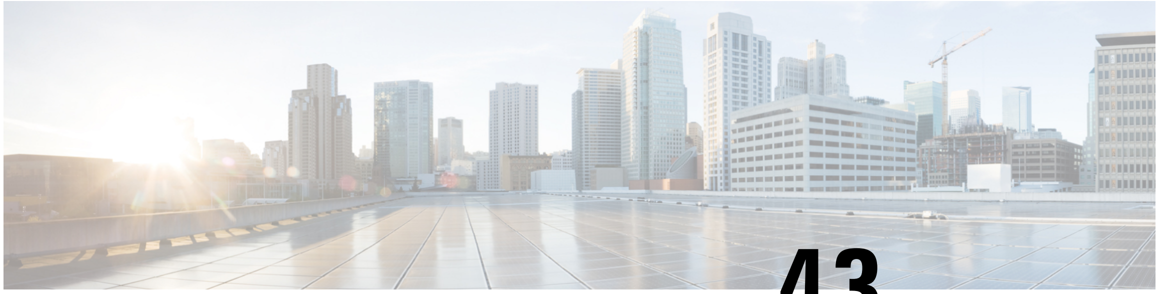
Table 59: History for Anonymous Reporting and Smart Call Home

Feature Name	Platform Releases	Description
Smart Call Home	8.2(2)	The Smart Call Home service offers proactive diagnostics and real-time alerts on the ASA, and provides higher network availability and increased operational efficiency. We introduced or modified the following commands: **active (call home)**, **call-home, call-home send alert-group, call-home test, contact-email-addr**, **customer-id (call home)**, **destination (call home)**, **profile**, **rename profile**, **service call-home, show call-home, show call-home detail**, **show smart-call-home alert-group**, **show call-home profile**, **show call-home statistics**, **show call-home mail-server status**, **show running-config call-home**, **show call-home registered-module status all**, **site-id**, **street-address**, **subscribe-to-alert-group all**, **alert-group-config**, **subscribe-to-alert-group configuration**, **subscribe-to-alert-group diagnostic**, **subscribe-to-alert-group environment**, **subscribe-to-alert-group inventory periodic**, **subscribe-to-alert-group snapshot periodic**, **subscribe-to-alert-group syslog**, **subscribe-to-alert-group telemetry periodic**.
Anonymous Reporting	9.0(1)	You can help to improve the ASA platform by enabling Anonymous Reporting, which allows Cisco to securely receive minimal error and health information from a device. We introduced the following commands: **call-home reporting anonymous**, **call-home test reporting anonymous**.

Feature Name	Platform Releases	Description
Smart Call Home	9.1(2)	The **show local-host** command was changed to the **show local-host | include interface** command for telemetry alert group reporting.
Smart Call Home	9.1(3)	A Smart Call Home message is sent to Cisco to report important cluster events if you have enabled clustering and configured Smart Call Home to subscribe to the Diagnostic alert group with a Critical severity level. A Smart Call Home clustering message is sent for only the following three events: • When a unit joins the cluster • When a unit leaves the cluster • When a cluster unit becomes the cluster control unit Each message that is sent includes the following information: • The active cluster member count • The output of the **show cluster info** command and the **show cluster history** command on the cluster control unit

PART IX

Reference



CHAPTER 43

Using the Command-Line Interface

This chapter describes how to use the CLI on the Cisco ASA.

Note The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the ASA operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works with or has the same function on the ASA.

Firewall Mode and Security Context Mode

The ASA runs in a combination of the following modes:

- Transparent firewall or routed firewall mode

  The firewall mode determines if the ASA runs as a Layer 2 or Layer 3 firewall.

- Multiple context or single context mode

  The security context mode determines if the ASA runs as a single device or as multiple security contexts, which act like virtual devices.

Some commands are only available in certain modes.

Command Modes and Prompts

The ASA CLI includes command modes. Some commands can only be entered in certain modes. For example, to enter commands that show sensitive information, you need to enter a password and enter a more privileged mode. Then, to ensure that configuration changes are not entered accidentally, you have to enter a configuration mode. All lower commands can be entered in higher modes, for example, you can enter a privileged EXEC command in global configuration mode.

Note The various types of prompts are all default prompts and when configured, they can be different.

- When you are in the system configuration or in single context mode, the prompt begins with the hostname:

  ```
  ciscoasa
  ```

- When printing the prompt string, the prompt configuration is parsed and the configured keyword values are printed in the order in which you have set the prompt command. The keyword arguments can be any of the following and in any order: hostname, domain, context, priority, state.

  prompt hostname context priority state

- When you are within a context, the prompt begins with the hostname followed by the context name:

  ```
  ciscoasa/context
  ```

The prompt changes depending on the access mode:

- User EXEC mode

  User EXEC mode lets you see minimum ASA settings. The user EXEC mode prompt appears as follows when you first access the ASA:

  ```
  ciscoasa>

  ciscoasa/context>
  ```

- Privileged EXEC mode

  Privileged EXEC mode lets you see all current settings up to your privilege level. Any user EXEC mode command will work in privileged EXEC mode. Enter the **enable** command in user EXEC mode, which requires a password, to start privileged EXEC mode. The prompt includes the number sign (#):

  ```
  ciscoasa#

  ciscoasa/context#
  ```

- Global configuration mode

Global configuration mode lets you change the ASA configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. Enter the **configure terminal** command in privileged EXEC mode to start global configuration mode. The prompt changes to the following:

```
ciscoasa(config)#
```

```
ciscoasa/context(config)#
```

- Command-specific configuration modes

  From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. For example, the **interface** command enters interface configuration mode. The prompt changes to the following:

```
ciscoasa(config-if)#
```

```
ciscoasa/context(config-if)#
```

Syntax Formatting

Command syntax descriptions use the conventions listed in the following table.

Table 60: Syntax Conventions

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical bar indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Abbreviate Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **wr t** to view the configuration instead of entering the full command **write terminal**, or you can enter **en** to start privileged mode and **conf t** to start configuration mode. In addition, you can enter **0** to represent **0.0.0.0**.

Command-Line Editing

The ASA uses the same command-line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

The ASA permits up to 512 characters in a command; additional characters are ignored.

Command Completion

To complete a command or keyword after entering a partial string, press the **Tab** key. The ASA only completes the command or keyword if the partial string matches only one command or keyword. For example, if you enter **s** and press the **Tab** key, the ASA does not complete the command because it matches more than one command. However, if you enter **dis**, the **Tab** key completes the **disable** command.

Command Help

Help information is available from the command line by entering the following commands:

- **help** *command_name*

  Shows help for the specific command.

- *command_name* **?**

  Shows a list of arguments available.

- *string***?** (no space)

  Lists the possible commands that start with the string.

- **?** and **+?**

  Lists all commands available. If you enter **?**, the ASA shows only commands available for the current mode. To show all commands available, including those for lower modes, enter **+?**.

Note If you want to include a question mark (?) in a command string, you must press **Ctrl-V** before typing the question mark so that you do not inadvertently invoke CLI help.

View the Running Configuration

To view the running configuration, use one of the following commands:

- **show running-config** [**all**] [*command*]

  If you specify **all**, then all default settings are shown as well. If you specify a *command*, then the output only includes related commands.

  Note Many passwords are shown as *****. To view the passwords in plain text, or in encrypted form if you have a master passphrase enabled, use the **more** command.

- **more system:running-config**

Filter show and more Command Output

You can use the vertical bar (|) with any **show** command and include a filter option and filtering expression. The filtering is performed by matching each output line with a regular expression, similar to Cisco IOS software. By selecting different filter options you can include or exclude all output that matches the expression. You can also display all output beginning with the line that matches the expression.

The syntax for using filtering options with the **show** command is as follows:

show *command* | {**include**| **exclude** | **begin** | **grep** [**-v**]} *regexp*

or

more system:running-config| {**include**| **exclude** | **begin** | **grep** [**-v**]} *regexp*

Note Entering the **more** command allows you to view the contents of any file, not just the running configuration; see the command reference for more information.

In this command string, the first vertical bar (|) is the operator and must be included in the command. This operator directs the output of the **show** command to the filter. In the syntax diagram, the other vertical bars (|) indicate alternative options and are not part of the command.

The **include** option includes all output lines that match the regular expression. The **grep** option without **-v** has the same effect. The **exclude** option excludes all output lines that match the regular expression. The **grep** option with**-v** has the same effect. The **begin** option shows all the output lines starting with the line that matches the regular expression.

Replace *regexp* with any Cisco IOS regular expression. The regular expression is not enclosed in quotes or double-quotes, so be careful with trailing white spaces, which will be taken as part of the regular expression.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters called *metacharacters* have special meaning when used in regular expressions.

Use **Ctrl+V** to escape all of the special characters in the CLI, such as a question mark (?) or a tab. For example, type **d[Ctrl+V]?g** to enter **d?g** in the configuration.

Redirecting and Appending show Command Output

Instead of displaying the output of a **show** command on the screen, you can redirect it to a file on the device or in a remote location. When redirecting to a file on the device, you can also append the command output to the file.

show *command* | {**append | redirect**} *url*

- **append** *url* adds the output to an existing file. Specify the file using one of the following:
    - **disk0:/**[[*path/*]*filename*] or **flash:/**[[*path/*]*filename*]—Both **flash** and **disk0** indicates the internal Flash memory. You can use either option.
    - **disk1:/**[[*path/*]*filename*]—Indicates external memory.
- **redirect** *url* creates the specified file, or overwrites it if the file already exists.
    - **disk0:/**[[*path/*]*filename*] or **flash:/**[[*path/*]*filename*]—Both **flash** and **disk0** indicates the internal Flash memory. You can use either option.
    - **disk1:/**[[*path/*]*filename*]—Indicates external memory.
    - **smb:/**[[*path/*]*filename*]—Indicates Server Message Block, a UNIX server local file system.
    - **ftp:**//[[*user*[:*password*]@] *server*[:*port*]/[*path/*] *filename*[**;type=***xx*]]—Indicates an SCP server. The **type** can be one of these keywords: **ap** (ASCII passive mode), **an** (ASCII normal mode), **ip** (Default—Binary passive mode), **in** (Binary normal mode).
    - **scp:**//[[*user*[:*password*]@] *server*[/*path*]/*filename*[**;int=***interface_name*]]—The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the Secure Copy (SCP) server.
    - **tftp:**//[[*user*[**:***password*]@] *server*[:*port*] /[*path/*]*filename*[**;int=***interface_name*]]—Indicates a TFTP server. The pathname cannot contain spaces. The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the TFTP server.

Command Output Paging

For commands such as **help** or **?**, **show**, **show xlate**, or other commands that provide long listings, you can determine if the information displays a screen and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- Press the **Space** bar to view another screen.
- Press the **Enter** key to view the next line.
- Press the q key to return to the command line.

Add Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

Text Configuration Files

This section describes how to format a text configuration file that you can download to the ASA.

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide.

In examples, commands are preceded by a CLI prompt. The prompt in the following example is “ciscoasa(config)#”:

```
ciscoasa(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted:

```
context a
```

Command-Specific Configuration Mode Commands

Command-specific configuration mode commands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the commands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
interface gigabitethernet0/0
nameif inside
interface gigabitethernet0/1
    nameif outside
```

Automatic Text Entries

When you download a configuration to the ASA, it inserts some lines automatically. For example, the ASA inserts lines for default settings or for the time the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as ACEs, are processed in the order they appear, and the order can affect the function of the access list. Other commands might also have order requirements. For example, you must enter the **nameif** command for an interface first because many subsequent commands use the name of the interface. Also, commands in a command-specific configuration mode must directly follow the main command.

Commands Not Included in the Text Configuration

Some commands do not insert lines in the configuration. For example, a runtime command such as **show running-config** does not have a corresponding line in the text file.

Passwords

The login, enable, and user passwords are automatically encrypted before they are stored in the configuration. For example, the encrypted form of the password "cisco" might look like jMorNbK0514fadBh. You can copy the configuration passwords to another ASA in its encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the ASA does not automatically encrypt it when you copy the configuration to the ASA. The ASA only encrypts it when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.

Multiple Security Context Files

For multiple security contexts, the entire configuration consists of the following multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the ASA, including a list of contexts
- The admin context, which provides network interfaces for the system configuration

  The system configuration does not include any interfaces or network settings for itself. Rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses a context that is designated as the admin context.

Each context is similar to a single context mode configuration. The system configuration differs from a context configuration in that the system configuration includes system-only commands (such as a list of all contexts) while other typical commands are not present (such as many interface parameters).

Supported Character Sets

The ASA CLI currently supports UTF-8 encoding only. UTF-8 is the particular encoding scheme for Unicode symbols, and has been designed to be compatible with an ASCII subset of symbols. ASCII characters are represented in UTF-8 as one-byte characters. All other characters are represented in UTF-8 as multibyte symbols.

The ASCII printable characters (0x20 to 0x7e) are fully supported. The printable ASCII characters are the same as ISO 8859-1. UTF-8 is a superset of ISO 8859-1, so the first 256 characters (0-255) are the same as ISO 8859-1. The ASA CLI supports up to 255 characters (multibyte characters) of ISO 8859-1.

CHAPTER 44

Addresses, Protocols, and Ports

This chapter provides a quick reference for IP addresses, protocols, and applications.

IPv4 Addresses and Subnet Masks

This section describes how to use IPv4 addresses in the Cisco ASA. An IPv4 address is a 32-bit number written in dotted-decimal notation: four 8-bit fields (octets) converted from binary to decimal numbers, separated by dots. The first part of an IP address identifies the network on which the host resides, while the second part identifies the particular host on the given network. The network number field is called the network prefix. All hosts on a given network share the same network prefix but must have a unique host number. In classful IP, the class of the address determines the boundary between the network prefix and the host number.

Classes

IP host addresses are divided into three different address classes: Class A, Class B, and Class C. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. Class D addresses are reserved for multicast IP.

- Class A addresses (1.xxx.xxx.xxx through 126.xxx.xxx.xxx) use only the first octet as the network prefix.
- Class B addresses (128.0.xxx.xxx through 191.255.xxx.xxx) use the first two octets as the network prefix.
- Class C addresses (192.0.0.xxx through 223.255.255.xxx) use the first three octets as the network prefix.

Because Class A addresses have 16,777,214 host addresses, and Class B addresses 65,534 hosts, you can use subnet masking to break these huge networks into smaller subnets.

Private Networks

If you need large numbers of addresses on your network, and they do not need to be routed on the Internet, you can use private IP addresses that the Internet Assigned Numbers Authority (IANA) recommends (see RFC 1918). The following address ranges are designated as private networks that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

Subnet Masks

A subnet mask lets you convert a single Class A, B, or C network into multiple networks. With a subnet mask, you can create an extended network prefix that adds bits from the host number to the network prefix. For example, a Class C network prefix always consists of the first three octets of the IP address. But a Class C extended network prefix uses part of the fourth octet as well.

Subnet masking is easy to understand if you use binary notation instead of dotted decimal. The bits in the subnet mask have a one-to-one correspondence with the Internet address:

- The bits are set to 1 if the corresponding bit in the IP address is part of the extended network prefix.
- The bits are set to 0 if the bit is part of the host number.

Example 1: If you have the Class B address 129.10.0.0 and you want to use the entire third octet as part of the extended network prefix instead of the host number, then you must specify a subnet mask of 11111111.11111111.11111111.00000000. This subnet mask converts the Class B address into the equivalent of a Class C address, where the host number consists of the last octet only.

Example 2: If you want to use only part of the third octet for the extended network prefix, then you must specify a subnet mask like 11111111.11111111.11111000.00000000, which uses only 5 bits of the third octet for the extended network prefix.

You can write a subnet mask as a dotted-decimal mask or as a /*bits* ("slash *bits*") mask. In Example 1, for a dotted-decimal mask, you convert each binary octet into a decimal number: 255.255.255.0. For a /*bits* mask, you add the number of 1s: /24. In Example 2, the decimal number is 255.255.248.0 and the /bits is /21.

You can also supernet multiple Class C networks into a larger network by using part of the third octet for the extended network prefix. For example, 192.168.0.0/20.

Determine the Subnet Mask

See the following table to determine the subnet mask based on how many hosts you want.

Note The first and last number of a subnet are reserved, except for /32, which identifies a single host.

Table 61: Hosts, Bits, and Dotted-Decimal Masks

Hosts	/Bits Mask	Dotted-Decimal Mask
16,777,216	/8	255.0.0.0 Class A Network
65,536	/16	255.255.0.0 Class B Network
32,768	/17	255.255.128.0
16,384	/18	255.255.192.0
8192	/19	255.255.224.0
4096	/20	255.255.240.0
2048	/21	255.255.248.0
1024	/22	255.255.252.0
512	/23	255.255.254.0
256	/24	255.255.255.0 Class C Network
128	/25	255.255.255.128
64	/26	255.255.255.192
32	/27	255.255.255.224
16	/28	255.255.255.240
8	/29	255.255.255.248
4	/30	255.255.255.252
Do not use	/31	255.255.255.254
1	/32	255.255.255.255 Single Host Address

Determine the Address to Use with the Subnet Mask

The following sections describe how to determine the network address to use with a subnet mask for a Class C-size and a Class B-size network.

Class C-Size Network Address

For a network between 2 and 254 hosts, the fourth octet falls on a multiple of the number of host addresses, starting with 0. For example, The following table shows the 8-host subnets (/29) of 192.168.0.x.

Note The first and last address of a subnet are reserved. In the first subnet example, you cannot use 192.168.0.0 or 192.168.0.7.

Table 62: Class C-Size Network Address

Subnet with Mask /29 (255.255.255.248)	Address Range
192.168.0.0	192.168.0.0 to 192.168.0.7
192.168.0.8	192.168.0.8 to 192.168.0.15
192.168.0.16	192.168.0.16 to 192.168.0.31
—	—
192.168.0.248	192.168.0.248 to 192.168.0.255

Class B-Size Network Address

To determine the network address to use with the subnet mask for a network with between 254 and 65,534 hosts, you need to determine the value of the third octet for each possible extended network prefix. For example, you might want to subnet an address like 10.1.*x*.0, where the first two octets are fixed because they are used in the extended network prefix, and the fourth octet is 0 because all bits are used for the host number.

To determine the value of the third octet, follow these steps:

1. Calculate how many subnets you can make from the network by dividing 65,536 (the total number of addresses using the third and fourth octet) by the number of host addresses you want.

   For example, 65,536 divided by 4096 hosts equals 16. Therefore, there are 16 subnets of 4096 addresses each in a Class B-size network.

2. Determine the multiple of the third octet value by dividing 256 (the number of values for the third octet) by the number of subnets:

   In this example, 256/16 = 16.

   The third octet falls on a multiple of 16, starting with 0.

The following table shows the 16 subnets of the network 10.1.

Note The first and last address of a subnet are reserved. In the first subnet example, you cannot use 10.1.0.0 or 10.1.15.255.

Table 63: Subnets of Network

Subnet with Mask /20 (255.255.240.0)	Address Range
10.1.0.0	10.1.0.0 to 10.1.15.255
10.1.16.0	10.1.16.0 to 10.1.31.255
10.1.32.0	10.1.32.0 to 10.1.47.255
—	—
10.1.240.0	10.1.240.0 to 10.1.255.255

IPv6 Addresses

IPv6 is the next generation of the Internet Protocol after IPv4. It provides an expanded address space, a simplified header format, improved support for extensions and options, flow labeling capability, and authentication and privacy capabilities. IPv6 is described in RFC 2460. The IPv6 addressing architecture is described in RFC 3513.

This section describes the IPv6 address format and architecture.

IPv6 Address Format

IPv6 addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. The following are two examples of IPv6 addresses:

- 2001:0DB8:7654:3210:FEDC:BA98:7654:3210
- 2001:0DB8:0000:0000:0008:0800:200C:417A

Note The hexadecimal letters in IPv6 addresses are not case-sensitive.

You do not need to include the leading zeros in an individual field of the address, but each field must contain at least one digit. So the example address 2001:0DB8:0000:0000:0008:0800:200C:417A can be shortened to 2001:0DB8:0:0:8:800:200C:417A by removing the leading zeros from the third through sixth fields from the left. The fields that contained all zeros (the third and fourth fields from the left) were shortened to a single zero. The fifth field from the left had the three leading zeros removed, leaving a single 8 in that field, and the sixth field from the left had the one leading zero removed, leaving 800 in that field.

It is common for IPv6 addresses to contain several consecutive hexadecimal fields of zeros. You can use two colons (::) to compress consecutive fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent the successive hexadecimal fields of zeros). The following table shows several examples of address compression for different types of IPv6 address.

Table 64: IPv6 Address Compression Examples

Address Type	Standard Form	Compressed Form
Unicast	2001:0DB8:0:0:0:BA98:0:3210	2001:0DB8::BA98:0:3210
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

Note Two colons (::) can be used only once in an IPv6 address to represent successive fields of zeros.

An alternative form of the IPv6 format is often used when dealing with an environment that contains both IPv4 and IPv6 addresses. This alternative has the format x:x:x:x:x:x:y.y.y.y, where x represent the hexadecimal values for the six high-order parts of the IPv6 address and y represent decimal values for the 32-bit IPv4 part of the address (which takes the place of the remaining two 16-bit parts of the IPv6 address). For example, the IPv4 address 192.168.1.1 could be represented as the IPv6 address 0:0:0:0:0:0:FFFF:192.168.1.1 or ::FFFF:192.168.1.1.

IPv6 Address Types

The following are the three main types of IPv6 addresses:

- **Unicast**—A unicast address is an identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. An interface may have more than one unicast address assigned to it.
- **Multicast**—A multicast address is an identifier for a set of interfaces. A packet sent to a multicast address is delivered to all addresses identified by that address.
- **Anycast**—An anycast address is an identifier for a set of interfaces. Unlike a multicast address, a packet sent to an anycast address is only delivered to the “nearest” interface, as determined by the measure of distances for the routing protocol.

Note There are no broadcast addresses in IPv6. Multicast addresses provide the broadcast functionality.

Unicast Addresses

This section describes IPv6 unicast addresses. Unicast addresses identify an interface on a network node.

Global Address

The general format of an IPv6 global unicast address is a global routing prefix followed by a subnet ID followed by an interface ID. The global routing prefix can be any prefix not reserved by another IPv6 address type.

All global unicast addresses, other than those that start with binary 000, have a 64-bit interface ID in the Modified EUI-64 format.

Global unicast address that start with the binary 000 do not have any constraints on the size or structure of the interface ID portion of the address. One example of this type of address is an IPv6 address with an embedded IPv4 address.

Site-Local Address

Site-local addresses are used for addressing within a site. They can be used to address an entire site without using a globally unique prefix. Site-local addresses have the prefix FEC0::/10, followed by a 54-bit subnet ID, and end with a 64-bit interface ID in the modified EUI-64 format.

Site-local routers do not forward any packets that have a site-local address for a source or destination outside of the site. Therefore, site-local addresses can be considered private addresses.

Link-Local Address

All interfaces are required to have at least one link-local address. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 and the interface identifier in modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes with a link-local address can communicate; they do not need a site-local or globally unique address to communicate.

Routers do not forward any packets that have a link-local address for a source or destination. Therefore, link-local addresses can be considered private addresses.

IPv4-Compatible IPv6 Addresses

There are two types of IPv6 addresses that can contain IPv4 addresses.

The first type is the IPv4-compatibly IPv6 address. The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that use this technique are assigned special IPv6 unicast addresses that carry a global IPv4 address in the low-order 32 bits. This type of address is termed an IPv4-compatible IPv6 address and has the format ::y.y.y.y, where y.y.y.y is an IPv4 unicast address.

Note The IPv4 address used in the IPv4-compatible IPv6 address must be a globally unique IPv4 unicast address.

The second type of IPv6 address, which holds an embedded IPv4 address, is called the IPv4-mapped IPv6 address. This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses. This type of address has the format ::FFFF:y.y.y.y, where y.y.y.y is an IPv4 unicast address.

Unspecified Address

The unspecified address, 0:0:0:0:0:0:0:0, indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

Note The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

Loopback Address

The loopback address, 0:0:0:0:0:0:0:1, may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

Note The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

Interface Identifiers

Interface identifiers in IPv6 unicast addresses are used to identify the interfaces on a link. They need to be unique within a subnet prefix. In many cases, the interface identifier is derived from the interface link-layer address. The same interface identifier may be used on multiple interfaces of a single node, as long as those interfaces are attached to different subnets.

For all unicast addresses, except those that start with the binary 000, the interface identifier is required to be 64 bits long and to be constructed in the Modified EUI-64 format. The Modified EUI-64 format is created from the 48-bit MAC address by inverting the universal/local bit in the address and by inserting the hexadecimal number FFFE between the upper three bytes and lower three bytes of the of the MAC address.

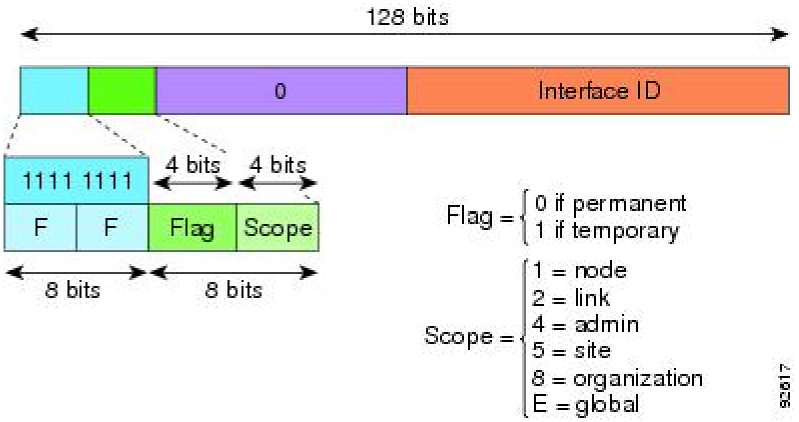
For example, and interface with the MAC address of 00E0.b601.3B7A would have a 64-bit interface ID of 02E0:B6FF:FE01:3B7A.

Multicast Address

An IPv6 multicast address is an identifier for a group of interfaces, typically on different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. An interface may belong to any number of multicast groups.

An IPv6 multicast address has a prefix of FF00::/8 (1111 1111). The octet following the prefix defines the type and scope of the multicast address. A permanently assigned (well known) multicast address has a flag parameter equal to 0; a temporary (transient) multicast address has a flag parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. The following figure shows the format of the IPv6 multicast address.

Figure 64: IPv6 Multicast Address Format



IPv6 nodes (hosts and routers) are required to join the following multicast groups:

- The All Nodes multicast addresses:
  - FF01:: (interface-local)
  - FF02:: (link-local)
- The Solicited-Node Address for each IPv6 unicast and anycast address on the node: FF02:0:0:0:0:1:FFXX:XXXX/104, where XX:XXXX is the low-order 24-bits of the unicast or anycast address.

Note Solicited-Node addresses are used in Neighbor Solicitation messages.

IPv6 routers are required to join the following multicast groups:

- FF01::2 (interface-local)
- FF02::2 (link-local)
- FF05::2 (site-local)

Multicast address should not be used as source addresses in IPv6 packets.

Note There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

Anycast Address

The IPv6 anycast address is a unicast address that is assigned to more than one interface (typically belonging to different nodes). A packet that is routed to an anycast address is routed to the nearest interface having that address, the nearness being determined by the routing protocol in effect.

Anycast addresses are allocated from the unicast address space. An anycast address is simply a unicast address that has been assigned to more than one interface, and the interfaces must be configured to recognize the address as an anycast address.

The following restrictions apply to anycast addresses:

- An anycast address cannot be used as the source address for an IPv6 packet.
- An anycast address cannot be assigned to an IPv6 host; it can only be assigned to an IPv6 router.

Note Anycast addresses are not supported on the ASA.

Required Addresses

IPv6 hosts must, at a minimum, be configured with the following addresses (either automatically or manually):

- A link-local address for each interface
- The loopback address
- The All-Nodes multicast addresses
- A Solicited-Node multicast address for each unicast or anycast address

IPv6 routers must, at a minimum, be configured with the following addresses (either automatically or manually):

- The required host addresses
- The Subnet-Router anycast addresses for all interfaces for which it is configured to act as a router

- The All-Routers multicast addresses

IPv6 Address Prefixes

An IPv6 address prefix, in the format ipv6-prefix/prefix-length, can be used to represent bit-wise contiguous blocks of the entire address space. The IPv6-prefix must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

The IPv6 prefix identifies the type of IPv6 address. The following table shows the prefixes for each IPv6 address type.

Table 65: IPv6 Address Type Prefixes

Address Type	Binary Prefix	IPv6 Notation
Unspecified	000...0 (128 bits)	::/128
Loopback	000...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local (unicast)	1111111010	FE80::/10
Site-Local (unicast)	1111111111	FEC0::/10
Global (unicast)	All other addresses.	
Anycast	Taken from the unicast address space.	

Protocols and Applications

The following table lists the protocol literal values and port numbers; either can be entered in ASA commands.

Table 66: Protocol Literal Values

Literal	Value	Description
ah	51	Authentication Header for IPv6, RFC 1826.
eigrp	88	Enhanced Interior Gateway Routing Protocol.
esp	50	Encapsulated Security Payload for IPv6, RFC 1827.
gre	47	Generic Routing Encapsulation.
icmp	1	Internet Control Message Protocol, RFC 792.
icmp6	58	Internet Control Message Protocol for IPv6, RFC 2463.
igmp	2	Internet Group Management Protocol, RFC 1112.

Literal	Value	Description
igrp	9	Interior Gateway Routing Protocol.
ip	0	Internet Protocol.
ipinip	4	IP-in-IP encapsulation.
ipsec	50	IP Security. Entering the ipsec protocol literal is equivalent to entering the esp protocol literal.
nos	94	Network Operating System (Novell's NetWare).
ospf	89	Open Shortest Path First routing protocol, RFC 1247.
pcp	108	Payload Compression Protocol.
pim	103	Protocol Independent Multicast.
pptp	47	Point-to-Point Tunneling Protocol. Entering the pptp protocol literal is equivalent to entering the gre protocol literal.
snp	109	Sitara Networks Protocol.
tcp	6	Transmission Control Protocol, RFC 793.
udp	17	User Datagram Protocol, RFC 768.

You can view protocol numbers online at the IANA website:

http://www.iana.org/assignments/protocol-numbers

TCP and UDP Ports

The following table lists the literal values and port numbers; either can be entered in ASA commands. See the following caveats:

- The ASA uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net. This value, however, does not agree with IANA port assignments.
- The ASA listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses the standard ports 1812 and 1813, you can configure the ASA to listen to those ports using the **authentication-port** and **accounting-port** commands.
- To assign a port for DNS access, use the **domain** literal value, not **dns**. If you use **dns**, the ASA assumes you meant to use the **dnsix** literal value.

You can view port numbers online at the IANA website:

http://www.iana.org/assignments/port-numbers

Table 67: Port Literal Values

Literal	TCP or UDP?	Value	Description
aol	TCP	5190	America Online
bgp	TCP	179	Border Gateway Protocol, RFC 1163
biff	UDP	512	Used by mail system to notify users that new mail is received
bootpc	UDP	68	Bootstrap Protocol Client
bootps	UDP	67	Bootstrap Protocol Server
chargen	TCP	19	Character Generator
cifs	TCP, UDP	3020	Common Internet File System
citrix-ica	TCP	1494	Citrix Independent Computing Architecture (ICA) protocol
cmd	TCP	514	Similar to exec except that cmd has automatic authentication
ctiqbe	TCP	2748	Computer Telephony Interface Quick Buffer Encoding
daytime	TCP	13	Day time, RFC 867
discard	TCP, UDP	9	Discard
dnsix	UDP	195	DNSIX Session Management Module Audit Redirector
domain	TCP, UDP	53	DNS
echo	TCP, UDP	7	Echo
exec	TCP	512	Remote process execution
finger	TCP	79	Finger
ftp	TCP	21	File Transfer Protocol (control port)
ftp-data	TCP	20	File Transfer Protocol (data port)
gopher	TCP	70	Gopher
h323	TCP	1720	H.323 call signaling
hostname	TCP	101	NIC Host Name Server
http	TCP, UDP	80	World Wide Web HTTP
https	TCP	443	HTTP over SSL
ident	TCP	113	Ident authentication service

Literal	TCP or UDP?	Value	Description
imap4	TCP	143	Internet Message Access Protocol, version 4
irc	TCP	194	Internet Relay Chat protocol
isakmp	UDP	500	Internet Security Association and Key Management Protocol
kerberos	TCP, UDP	750	Kerberos
klogin	TCP	543	KLOGIN
kshell	TCP	544	Korn Shell
ldap	TCP	389	Lightweight Directory Access Protocol
ldaps	TCP	636	Lightweight Directory Access Protocol (SSL)
login	TCP	513	Remote login
lotusnotes	TCP	1352	IBM Lotus Notes
lpd	TCP	515	Line Printer Daemon - printer spooler
mobile-ip	UDP	434	Mobile IP-Agent
nameserver	UDP	42	Host Name Server
netbios-dgm	UDP	138	NetBIOS Datagram Service
netbios-ns	UDP	137	NetBIOS Name Service
netbios-ssn	TCP	139	NetBIOS Session Service
nfs	TCP, UDP	2049	Network File System - Sun Microsystems
nntp	TCP	119	Network News Transfer Protocol
ntp	UDP	123	Network Time Protocol
pcanywhere-data	TCP	5631	pcAnywhere data
pcanywhere-status	UDP	5632	pcAnywhere status
pim-auto-rp	TCP, UDP	496	Protocol Independent Multicast, reverse path flooding, dense mode
pop2	TCP	109	Post Office Protocol - Version 2
pop3	TCP	110	Post Office Protocol - Version 3
pptp	TCP	1723	Point-to-Point Tunneling Protocol
radius	UDP	1645	Remote Authentication Dial-In User Service

Literal	TCP or UDP?	Value	Description
radius-acct	UDP	1646	Remote Authentication Dial-In User Service (accounting)
rip	UDP	520	Routing Information Protocol
rsh	TCP	514	Remote Shell
rtsp	TCP	554	Real Time Streaming Protocol
secureid-udp	UDP	5510	SecureID over UDP
sip	TCP, UDP	5060	Session Initiation Protocol
smtp	TCP	25	Simple Mail Transport Protocol
snmp	UDP	161	Simple Network Management Protocol
snmptrap	UDP	162	Simple Network Management Protocol - Trap
sqlnet	TCP	1521	Structured Query Language Network
ssh	TCP	22	Secure Shell
sunrpc	TCP, UDP	111	Sun Remote Procedure Call
syslog	UDP	514	System Log
tacacs	TCP, UDP	49	Terminal Access Controller Access Control System Plus
talk	TCP, UDP	517	Talk
telnet	TCP	23	RFC 854 Telnet
tftp	UDP	69	Trivial File Transfer Protocol
time	UDP	37	Time
uucp	TCP	540	UNIX-to-UNIX Copy Program
vxlan	UDP	4789	Virtual eXtensible Local Area Network (VXLAN)
who	UDP	513	Who
whois	TCP	43	Who Is
www	TCP, UDP	80	World Wide Web
xdmcp	UDP	177	X Display Manager Control Protocol

Local Ports and Protocols

The following table lists the protocols, TCP ports, and UDP ports that the ASA may open to process traffic destined to the ASA. Unless you enable the features and services listed in this table, the ASA does *not* open any local protocols or any TCP or UDP ports. You must configure a feature or service for the ASA to open the default listening protocol or port. In many cases you can configure ports other than the default port when you enable a feature or service.

Table 68: Protocols and Ports Opened by Features and Services

Feature or Service	Protocol	Port Number	Comments
DHCP	UDP	67,68	—
Failover Control	105	N/A	—
HTTP	TCP	80	—
HTTPS	TCP	443	—
ICMP	1	N/A	—
IGMP	2	N/A	Protocol only open on destination IP address 224.0.0.1
ISAKMP/IKE	UDP	500	Configurable.
IPsec (ESP)	50	N/A	—
IPsec over UDP (NAT-T)	UDP	4500	—
IPsec over TCP (CTCP)	TCP	—	No default port is used. You must specify the port number when configuring IPsec over TCP.
NTP	UDP	123	—
OSPF	89	N/A	Protocol only open on destination IP address 224.0.0.5 and 224.0.0.6
PIM	103	N/A	Protocol only open on destination IP address 224.0.0.13
RIP	UDP	520	—
RIPv2	UDP	520	Port only open on destination IP address 224.0.0.9
SNMP	UDP	161	Configurable.
SSH	TCP	22	—
Stateful Update	8 (non-secure) 9 (secure)	N/A	—

Feature or Service	Protocol	Port Number	Comments
Telnet	TCP	23	—
VPN Load Balancing	UDP	9023	Configurable.
VPN Individual User Authentication Proxy	UDP	1645, 1646	Port accessible only over VPN tunnel.

ICMP Types

The following table lists the ICMP type numbers and names that you can enter in ASA commands.

Table 69: ICMP Types

ICMP Number	ICMP Name
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
30	traceroute
31	conversion-error
32	mobile-redirect