Cisco Nexus 3000 Series NX-OS Security Configuration Guide, Release 9.3(x)

First Published: 2019-07-20

Last Modified: 2020-10-28

CONTENTS

CHAPTER 10   **Configuring Access Control Lists** **165**

Preface

This preface includes the following sections:

- Audience, on page xix
- Document Conventions, on page xix
- Related Documentation for Cisco Nexus 3000 Series Switches, on page xx
- Documentation Feedback, on page xx
- Communications, Services, and Additional Information, on page xx

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[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 set 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.

Convention	Description
variable	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
screen font	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/
tsd-products-support-series-home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

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.

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3000 Series NX-OS Security Guide, Release 9.3(x)*.

- New and Changed Information, on page 1

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3000 Series NX-OS Security Configuration Guide, Release 9.3(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.3(x)

Feature	Description	Changed in Release	Where Documented
NDcPP: OCSP for Syslog	Added OCSP support for syslog servers. This feature is supported on all Cisco Nexus 3000 Series switches.	9.3(5)	• Certificate Revocation Checking, on page 114 • Guidelines and Limitations for PKI, on page 115 • NDcPP: OCSP for Syslog, on page 115
ACL Consistency Checker	Added support for ACL consistency checker.	9.3(3)	ACL Consistency Checker, on page 174
No updates since Cisco NX-OS Release 9.2(x)	First 9.3(x) release	Not applicable	Not applicable

CHAPTER 2

Overview

The Cisco NX-OS software supports security features that can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Authentication, Authorization, and Accounting

Authentication, authorization, and accounting (AAA) is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

Authentication

Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.

Authorization

Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

Accounting

Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

Note   You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS or TACACS+, or if you want to configure a backup authentication method.

Related Topics

Configuring AAA

RADIUS and TACACS+ Security Protocols

AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS or TACACS+ security server.

The chapters in this guide describe how to configure the following security server protocols:

RADIUS

A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

TACACS+

A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

Related Topics

Configuring RADIUS

Configuring TACACS+

SSH and Telnet

You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

Related Topics

Configuring SSH and Telnet

IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Configuring IP ACLs

CHAPTER 3

Configuring FIPS

This chapter describes how to configure the Federal Information Processing Standards (FIPS) mode on Cisco NX-OS devices.

This chapter includes the following sections:

About FIPS

The FIPS 140–2 Publication, *Security Requirements for Cryptographic Modules*, details the U.S. government requirements for cryptographic modules. FIPS 140–2 specifies that a cryptographic module is a set of hardware, software, firmware, or some combination thereof that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary.

FIPS specifies certain cryptographic algorithms as secure, and it identifies which algorithms should be used if a cryptographic module is to be called FIPS compliant.

FIPS Self-Tests

A cryptographic module must perform power-up self-tests and conditional self-tests to ensure that it is functioning properly.

Power-up self-tests run automatically after the device powers up. A device goes into FIPS mode only after all self-tests are successfully completed. If any self-test fails, the device logs a system message and moves into an error state.

The device uses a cryptographic algorithm known-answer test (KAT) to test FIPS mode for each FIPS 140-2-approved cryptographic function (encryption, decryption, authentication, and random number generation)

Cisco Nexus 3000 Series NX-OS Security Configuration Guide, Release 9.3(x)

7

implemented on the device. The device applies the algorithm to data for which the correct output is already known. It then compares the calculated output to the previously generated output. If the calculated output does not equal the known answer, the KAT fails.

Conditional self-tests run automatically when an applicable security function or operation is invoked. Unlike the power-up self-tests, conditional self-tests are executed each time their associated function is accessed.

Conditional self-tests include the following:

Pair-wise consistency test

This test is run when a public or private key-pair is generated.

Continuous random number generator test

This test is run when a random number is generated.

The Cisco TrustSec manager also runs a bypass test to ensure that encrypted text is never sent as plain text.

> **Note** A bypass test failure on CTS-enabled ports causes only those corresponding ports to be shut down. The bypass test might fail because of packet drops caused by data path congestion. In such cases, we recommend that you try bringing up the port again.

FIPS Error State

When the system is booted up in FIPS mode, the FIPS power-up self-tests run on the supervisor and line card modules. If any of these bootup tests fail, the whole system is moved to the FIPS error state. In this state, as per the FIPS requirement, all cryptographic keys are deleted, and all line cards are shut down. This mode is exclusively meant for debugging purposes.

Once the switch is in the FIPS error state, any reload of a line card moves it to the failure state. To move the switch back to FIPS mode, it has to be rebooted. However, once the switch is in FIPS mode, any power-up self-test failure on a subsequent line card reload or insertion affects only that line card, and only the corresponding line card is moved to the failure state.

Prerequisites for FIPS

FIPS has the following prerequisites:

- Disable Telnet. Users should log in using Secure Shell (SSH) only.

- Disable SNMPv1 and v2. Any existing user accounts on the device that have been configured for SNMPv3 should be configured only with SHA for authentication and AES/3DES for privacy.

- Delete all SSH server RSA1 key-pairs.

- Enable HMAC-SHA1 message integrity checking (MIC) for use during the Cisco TrustSec Security Association Protocol (SAP) negotiation. To do so, enter the **sap hash-algorithm HMAC-SHA-1** command from the cts-manual or cts-dot1x mode.

Guidelines and Limitations for FIPS

FIPS has the following configuration guidelines and limitations:

- The user authentication mechanisms supported for SSH are usernames and passwords, public keys, and X.509 certificates.

- Your passwords should have a minimum of eight alphanumeric characters.

- Disable Radius and TACACS when FIPS mode is on. This is enforced due to OpenSSL in FIPS mode.

Default Settings for FIPS

This table lists the default settings for FIPS parameters.

Table 2: Default FIPS Parameters

Parameters	Default
FIPS mode	Disabled

Configuring FIPS

This section describes how to configure FIPS mode on Cisco NX-OS devices.

Enabling FIPs Mode

Beginning with Cisco NX-OS Release 7.0(3)I5(1), you can enable FIPS mode on the device.

Before you begin

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fips mode enable** **Example:** `switch(config)# fips mode enable`	Enables FIPS mode. **Note** **fips mode enable** can be entered only when all LCs are online or else it leads to LC failure.

		Command or Action	**Purpose**
Step 3		**exit**	Exits configuration mode.
		Example:	
		`switch(config)# exit` `switch#`	
Step 4		(Optional) **show fips status**	Displays the status of FIPS mode.
		Example:	
		`switch# show fips status` `FIPS mode is enabled`	
Step 5		Required: **copy running-config startup-config**	Copies the running configuration to the startup configuration.
		Example:	
		`switch# copy running-config` `startup-config`	
Step 6		Required: **reload**	Reloads the Cisco NX-OS device.
		Example:	**Note** After you enable FIPS, a reboot is required for the system to operate in FIPS mode.
		`switch# reload`	

Disabling FIPS

You can disable FIPS mode on the device.

Before you begin

Ensure that you are in the default VDC.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
	Example:	
	`switch# configure terminal` `switch(config)#`	
Step 2	**no fips mode enable**	Disables FIPS mode.
	Example:	
	`switch(config)# no fips mode enable`	
Step 3	**exit**	Exits configuration mode.
	Example:	
	`switch(config)# exit` `switch#`	

	Command or Action	Purpose

Step 4	(Optional) **show fips status**
	
	Example:
	`switch# show fips status`
	`FIPS mode is disabled`
		Displays the status of FIPS mode.

Step 5	**copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Step 6	**reload** **Example:** `switch# reload`	Reloads the Cisco NX-OS device.

Verifying the FIPS Configuration

To display FIPS configuration information, perform one of the following tasks:

Command	Purpose
show fips status	Displays the status of the FIPS feature.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 3000 Series NX-OS Security Command Reference*.

Create 2048 bit RSA Key

Steps to create a 2048 bit RSA key:

- `N9k-Switch# conf t`

  `Enter configuration commands, one per line. End with CNTL/Z.`

- `N9k-Switch(config)# no feature ssh`

  `XML interface to system may become unavailable since ssh is disabled`

- `N9k-Switch(config)# no ssh key rsa`

- `N9k-Switch(config)# ssh key rsa 2048`

- `New SSH Key has a bitcount of 2048:`

  `N9k-Switch(config)# show ssh key`

  `**************************************`

  `rsa Keys generated:Wed Apr 28 13:05:18 2021`

  `ssh-rsa   AAAAB3NzaC1yc2EAAAADAQABAAABAQDHpxEgZ9LwmbOEpJeJtLwqedmTLkZV7Setxb9D4xgO`

```
p2o2f6wt/48bPp/vLDGsxTF2PtLRtRSSDFNSQmkw9bg+MXvTpgNivdxWLjxtwo3YpYwPkBiReVmyrFgE

UuBmV/sDfhJpHXLoH9lR2+y0L5w1OG3cJxMe30TI37O3M8fZPjrAtHgkUubfEpiTbcyEw+aIHf+chyoR

eDJxcEdnlboiTDFR0/+jMUUM/vMtxd5x5DH3AO7htA/i8lvskrReR1CpX1sOOdcshmS57EEuEzR9cs+w

KSftQh6vLD802207T6+J7/+cXMVNQEbq0mCSzeTmOsuIQe8u9ZC24pgYzZ19

bitcount:2048

fingerprint:

SHA256:Am9861AIq5MzfSPQr4ZXGe0f5M9crnhk7HVZBXhMVBo

**************************************

could not retrieve dsa key information

**************************************

could not retrieve ecdsa key information

**************************************
```

Configuration Example for FIPS

The following example shows how to enable FIPS mode:

```
config terminal
fips mode enable
show fips status
exit
copy running-config startup-config
reload
```

Additional References for FIPS

This section includes additional information related to implementing FIPS.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 3000 Series NX-OS Security Command Reference*

Standards

Standards	Title
FIPS 140-2	Security Requirements for Cryptographic Modules

CHAPTER **4**

Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

Information About AAA

AAA Security Services

The authentication, authorization, and accounting (AAA) features allows you to verify the identity of, grant access to, and track the actions of users who manage Cisco Nexus devices. The Cisco Nexus device supports Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control device Plus (TACACS+) protocols.

Based on the user ID and password that you provide, the switches perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the switch and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- Authentication—Identifies users, including login and password dialog, challenge and response, messaging support, and, encryption depending on the security protocol that you select.

- Authorization—Provides access control.

  Authorization to access a Cisco Nexus device is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- Accounting—Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

> **Note** The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration

- Scalability

- Standardized authentication methods, such as RADIUS and TACACS+

- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch in the fabric are easier to manage.

- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.

- The accounting log for all switches in the fabric can be centrally managed.

- User attributes for each switch in the fabric are easier to manage than using the local databases on the switches.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. A server group provides for failover servers if a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, that server group option is considered a failure. If required, you can specify multiple server groups. If a switch encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

On Cisco Nexus devices, you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication

- Console login authentication

- User management session accounting

The following table lists the CLI commands for each AAA service configuration option.

Table 3: AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

- RADIUS server groups—Uses the global pool of RADIUS servers for authentication.

- Specified server groups—Uses specified RADIUS or TACACS+ server groups for authentication.

- Local—Uses the local username or password database for authentication.

- None—Uses only the username.

Note   If the method is for all RADIUS servers, instead of a specific server group, the Cisco Nexus devices choose the RADIUS server from the global pool of configured RADIUS servers in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco Nexus devices.

The following table describes the AAA authentication methods that you can configure for the AAA services.

Table 4: AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

Note   For console login authentication, user login authentication, and user management session accounting, the Cisco Nexus devices try each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Logins

The authentication and authorization process for user login is as occurs:

- When you log in to the required Cisco Nexus device, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.

- When you have configured the AAA server groups using the server group authentication method, the Cisco Nexus device sends an authentication request to the first AAA server in the group as follows:

  If the AAA server fails to respond, then the next AAA server is tried and so on until the remote server responds to the authentication request.

  If all AAA servers in the server group fail to respond, the servers in the next server group are tried.

  If all configured methods fail, the local database is used for authentication.

- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

  If the AAA server protocol is RADIUS, user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.

  If the AAA server protocol is TACACS+, another request is sent to the same server to get the user roles specified as custom attributes for the shell.

- If your username and password are successfully authenticated locally, the Cisco Nexus device logs you in and assigns you the roles configured in the local database.

The following figure shows a flowchart of the authentication and authorization process.

Figure 1: Authentication and Authorization Flow for User Login
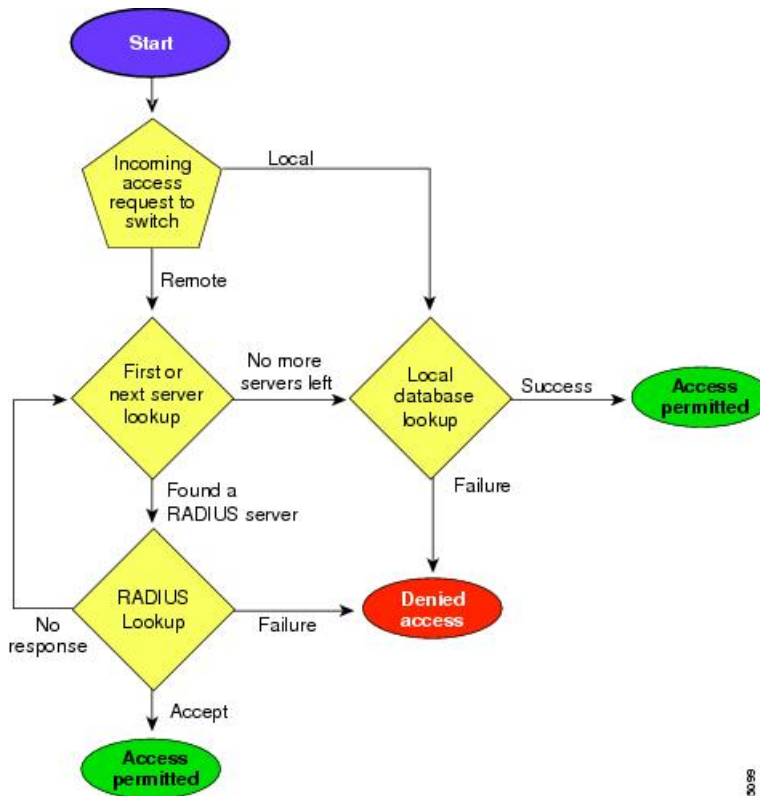
Note   This figure is applicable only to username password SSH authentication. It does not apply to public key SSH authentication. All username password SSH authentication goes through AAA.

In the figure, "No more servers left" means that there is no response from any server within this server group.

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable.

- The Cisco Nexus device is configured as a client of the AAA servers.

- The preshared secret key is configured on the Cisco Nexus device and on the remote AAA servers.

- The remote server responds to AAA requests from the Cisco Nexus device.

Guidelines and Limitations for AAA

The Cisco Nexus devices do not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally. If an all numeric username exists on an AAA server and is entered during a login, the Cisco Nexus device still logs in the user.

Caution   You should not create user accounts with usernames that are all numeric.

Configuring AAA

Configuring Console Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers

- Named subset of RADIUS or TACACS+ servers

- Local database on the Cisco Nexus device.

- Username only **none**

The default method is local.

✎

| Note | The **group radius** and **group** *server-name* forms of the **aaa authentication** command are used for a set of previously defined RADIUS servers. Use the **radius server-host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers. |

Before you configure console login authentication methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login console** {**group** *group-list* [**none**]	**local**	**none**}	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the console login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure authentication methods for the console login:

```
switch# configure terminal
switch(config)# aaa authentication login console group radius
switch(config)# exit
```

```
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The default method is local.

Before you configure default login authentication methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login default** {**group** *group-list* [**none**]	**local**	**none**}	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local** , which is used when no methods are configured or when all of the configured methods do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the default login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by the local user database if the remote AAA servers do not respond. If you have enabled the displaying of login failure messages, the following message is displayed:

```
Remote AAA servers unreachable; local authentication done.
Remote AAA servers unreachable; local authentication failed.
```

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login error-enable**	Enables login authentication failure messages. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the login failure message configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Logging Successful and Failed Login Attempts

You can configure the switch to log all successful and failed login attempts to the configured syslog server.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal	Enters global configuration mode.
Step 2	Required: [**no**] **login on-failure log** **Example:** switch(config)# **login on-failure log**	Logs all failed authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the failed login: AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.
Step 3	Required: [**no**] **login on-success log** **Example:** switch(config)# **login on-success log**	Logs all successful authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the successful login:

	Command or Action	Purpose
		AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message.
Step 4	(Optional) **show login on-failure log** **Example:** switch(config)# **show login on-failure log**	Displays whether the switch is configured to log failed authentication messages to the syslog server.
Step 5	(Optional) **show login on-successful log** **Example:** switch(config)# **show login on-successful log**	Displays whether the switch is configured to log successful authentication messages to the syslog server.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring AAA Command Authorization

When a TACACS+ server authorization method is configured, you can authorize every command that a user executes with the TACACS+ server which includes all EXEC mode commands and all configuration mode commands.

The authorization methods include the following:

- Group—TACACS+ server group

- Local—Local role-based authorization

- None—No authorization is performed

The default method is Local.

Note There is no authorization on the console session.

Before you begin

You must enable TACACS+ before configuring AAA command authorization.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**commands** | **config-commands**} {**default**} {{[**group** *group-name*] | [**local**]} | {[**group** *group-name*] | [**none**]}}
	
	Example:
	`switch(config)# aaa authorization`
	`config-commands default group tac1`
	
	Example:
	`switch# aaa authorization commands`
	`default group tac1`
		Configures authorization parameters.
		
		Use the **commands** keyword to authorize EXEC mode commandes.
		
		Use the **config-commands** keyword to authorize configuration mode commands.
		
		Use the **group**, **local**, or **none** keywords to identify the authorization method.

Example

The following example shows how to authorize EXEC mode commands with TACACS+ server group *tac1*:

`switch# `**`aaa authorization commands default group tac1`**

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

`switch(config)# `**`aaa authorization config-commands default group tac1`**

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, the command is authorized based on the user's *local* role.

`switch(config)# `**`aaa authorization config-commands default group tac1 local`**

The followng example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, allow the command regardless of the local role.

`switch# `**`aaa authorization commands default group tac1 none`**

The following example shows how to authorize EXEC mode commands regardless of the local role:

```
switch# aaa authorization commands default none
```

The following example shows how to authorize EXEC mode commands using the local role for authorization:

```
switch# aaa authorization commands default local
```

Enabling MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. You can use MSCHAP for user logins to a Cisco Nexus device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco Nexus device uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MSCHAP, you must configure your RADIUS server to recognize the MSCHAP vendor-specific attributes (VSAs).

The following table describes the RADIUS VSAs required for MSCHAP.

Table 5: MSCHAP RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP user in response to the challenge. It is only used in Access-Request packets.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login mschap enable**	Enables MS-CHAP authentication. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication login mschap**	Displays the MS-CHAP configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring AAA Accounting Default Methods

The Cisco Nexus device supports TACACS+ and RADIUS methods for accounting. The switches report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco Nexus device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:.

- RADIUS server group—Uses the global pool of RADIUS servers for accounting.

- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.

- Local—Uses the local username or password database for accounting.

$\begin{array}{c}\diagdown\\ \diagdown\end{array}$

Note   If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Before you configure AAA accounting default methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa accounting default** {**group** *group-list* | **local**}	Configures the default accounting method. One or more server group names can be specified in a space-separated list.
		The *group-list* argument consists of a space-delimited list of group names. The group names are the following:
		• **radius** —Uses the global pool of RADIUS servers for accounting.
		• *named-group* —Uses a named subset of TACACS+ or RADIUS servers for accounting.
		The **local** method uses the local database for accounting.
		The default method is **local**, which is used when no server groups are configured or when all the configured server group do not respond.

	Command or Action	**Purpose**
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa accounting**	Displays the configuration AAA accounting default methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Using AAA Server VSAs

VSAs

You can use vendor-specific attributes (VSAs) to specify the Cisco Nexus device user roles and SNMPv3 parameters on AAA servers.

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute seperator value *
```

The protocol is a Cisco attribute for a particular type of authorization, separator is an equal sign (=) for mandatory attributes, and an asterisk (*****) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco Nexus device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

VSA Format

The following VSA protocol options are supported by the Cisco Nexus device:

- Shell— Used in access-accept packets to provide user profile information.

- Accounting—Used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Cisco Nexus device:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space.

- accountinginfo—Stores additional accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco Nexus device using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

Note For information on Cisco Unified Wireless Network TACACS+ configurations and to change the user roles, see Cisco Unified Wireless Network TACACS+ Configuration.

You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

For additional information, see the Configuring User Accounts and RBAC chapter in the System Management Configuration Guide for your Cisco Nexus device.

Secure Login Enhancements

Secure Login Enhancements

The following secure login enhancements are supported in Cisco NX-OS:

- Configuring Login Parameters

- Configuration Examples for Login Parameters

- Restricting Sessions Per User—Per User Per Login

- Enabling the Password Prompt for User Name

- Configuring Share Key Value for using RADIUS/TACACS+

Configuring Login Parameters

Use this task to configure your Cisco NX-OS device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

All login parameters are disabled by default. You must enter the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following default is enforced:

- All login attempts made through Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is entered.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**[no] login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** Switch(config)# login block-for 100 attempts 2 within 100	Configures your Cisco NX-OS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.
Step 3	**[no] login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** Switch(config)# login quiet-mode access-class myacl	(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.
Step 4	**exit** **Example:** Switch(config)# exit	Exits to privileged EXEC mode.
Step 5	**show login failures** **Example:** Switch# show login	Displays login parameters. • **failures** --Displays information related only to failed login attempts.

Configuration Examples for Login Parameters

Setting Login Parameters Example

The following example shows how to configure your switch to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests are denied during the quiet period except hosts from the ACL "myacl."

```
Switch(config)# login block-for 100 attempts 15 within 100
Switch(config)# login quiet-mode access-class myacl
```

Showing Login Parameters Example

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Switch# show login

No Quiet-Mode access list has been configured, default ACL will be applied.

Switch is enabled to watch for login Attacks.
If more than 2 login failures occur in 45 seconds or less,  logins will be disabled for 70
 seconds.

Switch presently in Normal-Mode.
Current Watch Window remaining time 10 seconds.
Present login failure count 0.
```

The following sample output from the **show login failures** command shows all failed login attempts on the switch:

```
Switch# show login failures

Information about last 20 login failures with the device.
--------------------------------------------------------------------------------
Username                              Line    Source            Appname
TimeStamp
--------------------------------------------------------------------------------
admin                                pts/0   bgl-ads-728.cisco.com   login
        Wed Jun 10 04:56:16 2015
admin                                pts/0   bgl-ads-728.cisco.com   login
        Wed Jun 10 04:56:19 2015
--------------------------------------------------------------------------------
```

The following sample output from the **show login failures** command verifies that no information is presently logged:

```
Switch# show login failures
*** No logged failed login attempts with the device.***
```

About No Service Password-Recovery

The No Service Password-Recovery feature enables anyone with console access, the ability to access the router and its network. The No Service Password-Recovery feature prevents the password recovery with standard procedure as described in the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

Enabling No Service Password-Recovery

If the no service password-recovery feature is enabled, then none except the administrator with network privileges will be able to modify the administrator password.

Before you begin

If you plan to enter the no service password-recovery command, Cisco recommends that you save a copy of the system configuration file in a location away from the device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	**no service password-recovery** **Example:** switch(config)# no service password-recovery WARNING: Executing this command will disable the password recovery mechanism. Do not execute this command without another plan for password recovery. Are you sure you want to continue? (y/n) : [y] y switch(config)# copy run start [######################################] 100% Copy complete, now saving to disk (please wait)... Copy complete.	Disables the password recovery mechanism.
Step 3	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.
Step 4	**Reload** **Example:** switch(config)# Reload This command will reboot the system. (y/n)? [n] y 2018 Jun 26 16:23:19 BAR %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command Line Interface CISCO SWITCH Ver 8.34 CISCO SWITCH Ver 8.34 Manual system restart from Command Line Interface writing reset reason 9, ... switch(boot)# config t Enter configuration commands, one per line. End with CNTL/Z. switch(boot)(config)# admin-password Abcd!123$ ERROR: service password-recovery disabled. Cannot change password! switch(boot)(config)#	
Step 5	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	switch(config)# exit switch#	
Step 6	(Optional) **show user-account** **Example:** switch# show user-account	Displays the role configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Restricting Sessions Per User—Per User Per Login

Use this task to restrict the maximum sessions per user.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**[no] user max-logins** *max-logins* **Example:** Switch(config)# user max-logins 1	Restricts the maximum sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, then only one session (telnet/SSH) is allowed per user.
Step 3	**exit** **Example:** Switch(config)# exit	Exits to privileged EXEC mode.

Enabling the Password Prompt for User Name

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**[no] password prompt username** **Example:** Switch(config)# password prompt username	Enables the login knob. If this command is enabled and the user enters the **username** command without the password option, then the password is prompted. The password accepts hidden characters. Use the **no** form of this command to disable the login knob.
Step 3	**exit** **Example:** Switch(config)# exit	Exits to privileged EXEC mode.

Configuring Share Key Value for using RADIUS/TACACS+

The shared secret you configure for remote authentication and accounting must be hidden. For the **radius-server key** and **tacacs-server key** commands, a separate command to generate encrypted shared secret can be used.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**generate type7_encrypted_secret** **Example:** Switch(config)# generate type7_encrypted_secret	Configures RADIUS and TACACS shared secret with key type 7. While generating an encrypted shared secret, user input is hidden. **Note** You can generate encrypted equivalent of plain text separately and can configure the encrypted shared secret later.
Step 3	**exit** **Example:** Switch(config)# exit	Exits to privileged EXEC mode.

Monitoring and Clearing the Local AAA Accounting Log

The Cisco Nexus device maintains a local log for the AAA accounting activity.

Procedure

	Command or Action	Purpose
Step 1	switch# **show accounting log** [*size*] [**start-time** *year month day hh* **:** *mm* **:** *ss*]	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the size argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a start time for the log output.
Step 2	(Optional) switch# **clear accounting log**	Clears the accounting log contents.

Verifying the AAA Configuration

To display AAA configuration information, perform one of the following tasks:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
show aaa authorization	Displays AAA authorization information.
show aaa groups	Displays the AAA server group configuration.
show login [**failures**]	Displays the login parameters. The **failures** option displays information related only to failed login attempts. **Note** The **clear login failures** command clears the login failures in the current watch period.
show login on-failure log	Displays whether the switch is configured to log failed authentication messages to the syslog server.
show login on-successful log	Displays whether the switch is configured to log successful authentication messages to the syslog server.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show running-config all | **i max-login**	Displays the maximum number of login sessions allowed per user.

Command	Purpose
show startup-config aaa	Displays the AAA configuration in the startup configuration.
show userpassphrase {**length** | **max-length** | **min-length**}	Displays the minimum and maximum length of the user password.

Configuration Examples for AAA

The following example shows how to configure AAA:

```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

Default AAA Settings

The following table lists the default settings for AAA parameters.

Table 6: Default AAA Parameters

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled
MSCHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB

Configuring RADIUS

This chapter contains the following sections:

Information About RADIUS

The Remote Access Dial-In User Service (RADIUS) distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco Nexus devices and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS Network Environments

RADIUS can be implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS.

For example, network devices from several vendors can use a single RADIUS server-based security database.

- Networks already using RADIUS.

  You can add a Cisco Nexus device with RADIUS to the network. This action might be the first step when you make a transition to an AAA server.

- Networks that require resource accounting.

  You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.

- Networks that support authentication profiles.

  Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the Cisco Nexus device to manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

Information About RADIUS Operations

When a user attempts to log in and authenticate to a Cisco Nexus device using RADIUS, the following process occurs:

1. The user is prompted for and enters a username and password.

2. The username and encrypted password are sent over the network to the RADIUS server.

3. The user receives one of the following responses from the RADIUS server:

   - ACCEPT—The user is authenticated.

   - REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.

   - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.

   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.

- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause delay in processing of AAA requests. You can configure the switch to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The switch marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The switch periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This process verifies that a RADIUS server is in a working state before real AAA requests are sent to the server. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the switch displays an error message that a failure is taking place.

The following figure shows the different RADIUS server states:

Figure 2: RADIUS Server States
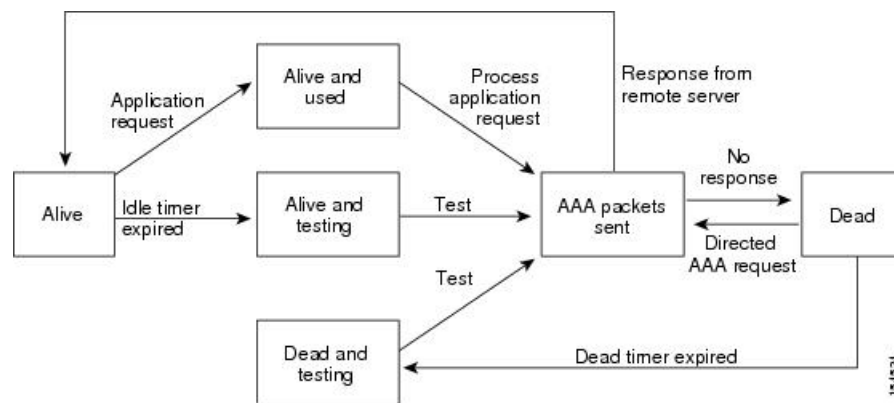
Note   The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is an equal sign (=) for mandatory attributes, and an asterisk (*) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco Nexus device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Cisco Nexus device:

- Shell— Used in access-accept packets to provide user profile information.

- Accounting— Used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Cisco Nexus device supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white spaces.

- accountinginfo—Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch. It can be used only with the accounting protocol data units (PDUs).

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco NX-OS sofware supports the RADIUS Change of Authorization (CoA) request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

When Dot1x is enabled, the network device acts as the authenticator and is responsible for processing dynamic COA per session.

The following requests are supported:

- Session reauthentication

- Session termination

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the response of the device to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the device responds by sending an Extensible Authentication Protocol over LAN (EAPOL)-RequestId message to the server.

- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.

- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network.

If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute.

If the session is located, but the NAS was unable to remove the session due to some internal error, the device returns a Disconnect-NAK message with the "Session Context Not Removable" error-code attribute.

If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- You must obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.

- You must obtain preshared keys from the RADIUS servers.

- Ensure that the Cisco Nexus device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following configuration guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco Nexus device.

Guidelines and Limitations for RADIUS Change of Authorization

RADIUS Change of Authorization has the following guidelines and limitations:

- RADIUS Change of Authorization is supported on FEX.

- RADIUS change of Authorization is supported for VXLAN EVPN.

Default Settings for RADIUS

The following table lists the default settings for RADIUS parameters.

Table 7: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

This section describes how to configure RADIUS servers.

Procedure

Step 1   Establish the RADIUS server connections to the Cisco Nexus device.

Step 2   Configure the preshared secret keys for the RADIUS servers.

Step 3   If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.

Step 4   If needed, configure any of the following optional parameters:

- Dead-time interval.

- Allow specification of a RADIUS server at login.

- Transmission retry count and timeout interval.

- Accounting and authentication attributes.

Step 5   If needed, configure periodic RADIUS server monitoring.

Configuring RADIUS Server Hosts

You must configure the IPv4 or IPv6 address or the hostname for each RADIUS server that you want to use for authentication. All RADIUS server hosts are added to the default RADIUS server group. You can configure up to 64 RADIUS servers.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address*	*ipv6-address*	*host-name*}	Specifies the IPv4 or IPv6 address or hostname for a RADIUS server.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure host 10.10.1.1 as a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1
switch(config)# exit
switch# copy running-config startup-config
```

Configuring RADIUS Global Preshared Keys

You can configure preshared keys at the global level for all servers used by the Cisco Nexus device. A preshared key is a shared secret text string between the switch and the RADIUS server hosts.

Before you begin

Obtain the preshared key values for the remote RADIUS servers

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server key** [**0**	**7**] *key-value*	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure preshared keys at the global level for all servers used by the device:

```
switch# configure terminal
switch(config)# radius-server key 0 QsEfThUkO
switch(config)# exit
switch# copy running-config startup-config
```

Configuring RADIUS Server Preshared Keys

A preshared key is a shared secret text string between the Cisco Nexus device and the RADIUS server host.

Before you begin

Obtain the preshared key values for the remote RADIUS servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS preshared keys:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring RADIUS Server Groups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters. To delete a RADIUS server group, use the **no** form of this command. **Note** You are not allowed to delete the default system generated default group (RADIUS).
Step 3	switch (config-radius)# **server** {*ipv4-address* | *ipv6-address* | *server-name*}	Configures the RADIUS server as a member of the RADIUS server group.

	Command or Action	Purpose
		If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) switch (config-radius)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440.
		Note If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-radius)# **source-interface** *interface*	Assigns a source interface for a specific RADIUS server group.
		The supported interface types are management and VLAN.
		Note Use the **source-interface** command to override the global source interface assigned by the **ip radius source-interface** command.
Step 6	switch(config-radius)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server group** [*group-name*]	Displays the RADIUS server group configuration.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure a RADIUS server group:

```
switch# configure terminal
switch (config)# aaa group server radius RadServer
switch (config-radius)# server 10.10.1.1
switch (config-radius)# deadtime 30
switch (config-radius)# use-vrf management
switch (config-radius)# exit
switch (config)# show radius-server group
switch (config)# copy running-config startup-config
```

What to do next

Apply the RADIUS server groups to an AAA service.

Configuring the Global Source Interface for RADIUS Server Groups

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. You can also configure a different source interface for a specific RADIUS server group.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip radius source-interface** *interface*	Configures the global source interface for all RADIUS server groups configured on the device. The source interface can be the management or the VLAN interface.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration information.
Step 5	(Optional) switch# **copy running-config startup config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the mgmt 0 interface as the global source interface for RADIUS server groups:

```
switch# configure terminal
switch(config)# ip radius source-interface mgmt 0
switch(config)# exit
switch# copy running-config startup-config
```

Allowing Users to Specify a RADIUS Server at Login

You can allow users to specify a RADIUS server at login.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server directed-request**	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) switch# **show radius-server directed-request**	Displays the directed request configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to allow users to select a RADIUS server when logging in to a network:

```
switch# configure terminal
switch(config)# radius-server directed-request
switch# exit
switch# copy running-config startup-config
```

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the Cisco Nexus device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server retransmit** *count*	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	switch(config)# **radius-server timeout** *seconds*	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set the retry count to 3 and the transmission timeout interval to 5 seconds for RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server retransmit 3
switch(config)# radius-server timeout 5
switch(config)# exit
switch# copy running-config startup-config
```

Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Server

By default, a Cisco Nexus switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the switch waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address*	*ipv6-address*	*host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.
Step 3	switch(config)#**radius-server host** {*ipv4-address*	*ipv6-address*	*host-name*} **timeout** *seconds*	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to set the RADIUS transmission retry count to 3 and the timeout interval to 10 seconds on RADIUS host server server1:

```
switch# configure terminal
switch(config)# radius-server host server1 retransmit 3
```

```
switch(config)# radius-server host server1 timeout 10
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*	Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**	Specifies that the specified RADIUS server is to be used only for accounting purposes. The default is both accounting and authentication.
Step 4	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port*	Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication**	Specifies that the specified RADIUS server only be used for authentication purposes. The default is both accounting and authentication.
Step 6	switch(config)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server**	Displays the RADIUS server configuration.
Step 8	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure accounting and authentication attributes for a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 acct-port 2004
switch(config)# radius-server host 10.10.1.1 accounting
switch(config)# radius-server host 10.10.2.2 auth-port 2005
switch(config)# radius-server host 10.10.2.2 authentication
switch # exit
```

```
switch # copy running-config startup-config
switch #
```

Configuring Periodic RADIUS Server Monitoring

You can monitor the availability of RADIUS servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval during which a RADIUS server receives no requests before the switch sends out a test packet. You can configure this option to test servers periodically.

Note   For security reasons, we recommend that you do not configure a test username that is the same as an existing user in the RADIUS database.

The test idle timer specifies the interval during which a RADIUS server receives no requests before the switch sends out a test packet.

The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the switch does not perform periodic RADIUS server monitoring.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	switch(config)# **radius-server deadtime** *minutes*	Specifies the number of minutes before the switch checks a RADIUS server that was previously unresponsive. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS server host 10.10.1.1 with a username (user1) and password (Ur2Gd2BH) and with an idle timer of 3 minutes and a deadtime of 5 minutes:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time
 3
switch(config)# radius-server deadtime 5
switch(config)# exit
switch# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco Nexus device waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server deadtime**	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a deadtime of 5 minutes for a radius server:

```
switch# configure terminal
switch(config)# radius-server deadtime 5
switch(config# exit
switch# copy running-config startup-config
```

Manually Monitoring RADIUS Servers or Groups

Procedure

	Command or Action	Purpose
Step 1	switch# **test aaa server radius** {*ipv4-address*| *ipv6-address* | *server-name*} [**vrf** *vrf-name*] *username password* **test aaa server radius** {*ipv4-address* | *ipv6-address* | *server-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a RADIUS server to confirm availability.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a RADIUS server group to confirm availability.

Example

This example shows how to send a test message to the RADIUS server and server group to confirm availability:

```
switch# test aaa server radius 10.10.1.1 user 1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Enabling or Disabling Dynamic Author Server

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**aaa server radius dynamic-author** **Example:** switch(config)# **aaa server radius dynamic-author**	Enables the RADIUS dynamic author server. You can disable the RADIUS dynamic author server using the no form of this command.

Configuring RADIUS Change of Authorization

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **aaa server radius dynamic-author**
	
	Example:
	`switch(config)# aaa server radius`
	`dynamic-author`
		Configures the switch as an AAA server to facilitate interaction with an external policy server. You can disable the RADIUS dynamic author and the associated clients using the no form of this command.

Step 3	[**no**] **client** {*ip-address* | **hostname** } [**server-key** [**0** | **7**] *string*]
	
	Example:
	`switch(config-locsvr-da-radius)# client`
	`192.168.0.5 server-key cisco1`
		Configures the IP address or the hostname of the AAA server client. Use the optional server-key keyword and string argument to configure the server key at the client level. You can remove the client server using the no form of this command.
		
		Note Configuring the server key at the client level overrides the server key that is configured at the global level.

Step 4	[**no**] **port** *port-number*
	
	Example:
	`switch(config-locsvr-da-radius)# port`
	`3799`
		Specifies the port on which a device listens to the RADIUS requests from the configured RADIUS clients. The port range is 1 - 65535. You can revert to the default port using the no form of this command.
		
		Note The default port for a packet of disconnect is 1700.

Step 5	[**no**] **server-key** [**0** | **7**] *string*	Configures the global RADIUS key to be shared between a device and the RADIUS clients. You can remove the server-key using the no form of this command.

Verifying RADIUS Change of Authorization Configuration

To display RADIUS Change of Authorization configuration information, perform one of the following tasks:

Command	Purpose
show running-config dot1x	Displays the dot1x configuration in the running configuration.

Command	Purpose
show running-config aaa	Displays the AAA configuration in the running configuration.
show running-config radius	Displays the RADIUS configuration in the running configuration.
show aaa server radius statistics	Displays the local RADIUS server statistics.
show aaa client radius statistics {*ip address* | *hostname* }	Displays the local RADIUS client statistics.
clear aaa server radius statistics	Clears the local RADIUS server statistics.
clear aaa client radius statistics {*ip address* | *hostname* }	Clears the local RADIUS client statistics.

Verifying the RADIUS Configuration

To display AAA information, perform one of the following tasks:

Command	Purpose
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*server-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

Displaying RADIUS Server Statistics

Procedure

	Command or Action	Purpose
Step 1	switch# **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}	Displays the RADIUS statistics.

Clearing RADIUS Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show radius-server statistics** {*hostname*	*ipv4-address*	*ipv6-address*}	Displays the RADIUS server statistics on the Cisco NX-OS device.
Step 2	switch# **clear radius-server statistics** {*hostname*	*ipv4-address*	*ipv6-address*}	Clears the RADIUS server statistics.

Configuration Examples for RADIUS

The following example shows how to configure RADIUS:

```
switch# configure terminal
switch(config)# radius-server key 7 "ToIkLhPpG"
switch(config)# radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# exit
switch(config-radius)# use-vrf management
```

Configuration Examples of RADIUS Change of Authorization

The following example shows how to configure RADIUS Change of Authorization:

```
radius-server host 10.77.143.170 key 7 "fewhg123" authentication accounting
aaa server radius dynamic-author
    client 10.77.143.170 vrf management server-key 7 "fewhg123"
```

Default Settings for RADIUS

The following table lists the default settings for RADIUS parameters.

Table 8: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1

Parameters	Default
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Feature History for RADIUS

Table 9: Feature History for RADIUS

Feature Name	Releases	Feature Information
RADIUS	5.0(3)U1(1)	This feature was introduced.
IPv6	5.0(3)U3(1)	IPv6 support was introduced.

Configuring TACACS+

This chapter contains the following sections:

Information About Configuring TACACS+

The Terminal Access Controller Access Control System Plus (TACACS+) security protocol provides centralized validation of users attempting to gain access to a Cisco Nexus device. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your Cisco Nexus device are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service (authentication, authorization, and accounting) independently. Each service is associated with its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. The Cisco Nexus device provides centralized authentication using the TACACS+ protocol.

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Nexus device can authorize access without authenticating.

- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.

• Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

User Login with TACACS+

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco Nexus device using TACACS+, the following actions occur:

1. When the Cisco Nexus device establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.

> **Note** TACACS+ allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination, but may include prompts for other items, such as the user's mother's maiden name.

2. The Cisco Nexus device receives one of the following responses from the TACACS+ daemon:

   • ACCEPT—User authentication succeeds and service begins. If the Cisco Nexus device requires user authorization, authorization begins.

   • REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.

   • ERROR—An error occurred at some time during authentication dither at the daemon or in the network connection between the daemon and the Cisco Nexus device. If the Cisco Nexus deviceh receives an ERROR response, the switch tries to use an alternative method for authenticating the user.

   The user also undergoes an additional authorization phase, if authorization has been enabled on the Cisco Nexus device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the Cisco Nexus device again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

   Services include the following:

   • • Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services

     • Connection parameters, including the host or client IP address (IPv4), access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

You must configure the TACACS+ that is preshared key to authenticate the switch to the TACACS+ server. A preshared key is a secret text string shared between the Cisco Nexus device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global preshared secret key for all TACACS+ server configurations on the Cisco Nexus deviceh to use.

You can override the global preshared key assignment by using the **key** option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco Nexus device can periodically monitor an TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco Nexus device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. The Cisco Nexus device periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent to the server. Whenever an TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco Nexus device displays an error message that a failure is taking place before it can impact performance.

The following figure shows the different TACACS+ server states:

Figure 3: TACACS+ Server States
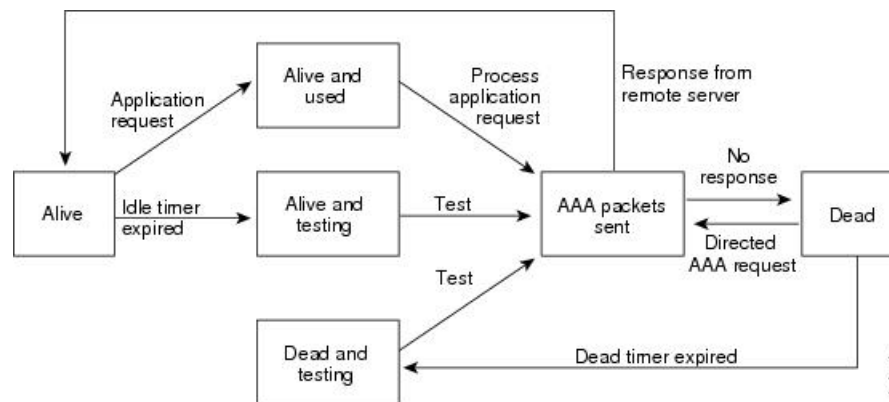
Note   The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- You must obtain the IPv4 addresses or hostnames for the TACACS+ servers.
- You must obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations for TACACS+

TACACS+ has the following configuration guidelines and limitations:

- You can configure a maximum of 64 TACACS+ servers on the Cisco Nexus device.

- You may get the following error message sporadically after you have configured a TACACS+ server host followed by the AAA configuration to actually use the host:

  ```
  %TACACS-3-TACACS_ERROR_MESSAGE: All servers failed to respond
  ```

  This is a known issue and there is no workaround. If the remote authentication works properly without any TACACS server connectivity issue, you can ignore the message and continue with your further configuration.

Configuring TACACS+

TACACS+ Server Configuration Process

This section describes how to configure TACACS+ servers.

Procedure

Step 1	Enable TACACS+.
Step 2	Establish the TACACS+ server connections to the Cisco Nexus device.
Step 3	Configure the preshared secret keys for the TACACS+ servers.
Step 4	If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.
Step 5	If needed, configure any of the following optional parameters:
	• Dead-time interval
	• Allow TACACS+ server specification at login
	• Timeout interval
	• TCP port
Step 6	If needed, configure periodic TACACS+ server monitoring.

Enabling TACACS+

Although by default, the TACACS+ feature is disabled on the Cisco Nexus device. You can enable the TACACS+ feature to access the configuration and verification commands for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature tacacs+**	Enables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IPv4 address or the hostname for the TACACS+ server on the Cisco Nexus device. All TACACS+ server hosts are added to the default TACACS+ server group.You can configure up to 64 TACACS+ servers.

If a preshared key is not configured for a configured TACACS+ server, a warning message is issued if a global key is not configured. If a TACACS+ server key is not configured, the global key (if configured) is used for that server.

Before you configure TACACS+ server hosts, you should do the following:

- Enable TACACS+.

- Obtain the IPv4 addresses or the hostnames for the remote TACACS+ servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **exit**	Exits configuration mode.
Step 3	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

You can delete a TACACS+ server host from a server group.

Configuring TACACS+ Global Preshared Keys

You can configure preshared keys at the global level for all servers used by the Cisco Nexus device. A preshared key is a shared secret text string between the Cisco Nexus device and the TACACS+ server hosts.

Before you configure preshared keys, you should do the following:

- Enable TACACS+.

• Obtain the preshared key values for the remote TACACS+ servers.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server key** [**0** | **6** | **7**] *key-value* **Example:** switch(config)# **tacacs-server key 0 QsEfThUkO** **Example:** switch(config)# **tacacs-server key 7 "fewhg"**	Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no secret key is configured. **Note** If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure global preshared keys:

```
switch# configure terminal
switch(config)# tacacs-server key 0 QsEfThUkO
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring TACACS+ Server Preshared Keys

You can configure preshared keys for a TACACS+ server. A preshared key is a shared secret text string between the Cisco Nexus device and the TACACS+ server host.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **exit**	Exits configuration mode.
Step 3	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure the TACACS+ preshared keys:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring TACACS+ Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the TACACS+ protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

You must use the **feature tacacs**+ command to enable TACACS+ before you configure TACACS+.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa group server tacacs**+ *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.

	Command or Action	Purpose
Step 3	(Optional) switch(config-tacacs+)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 4	(Optional) switch(config-tacacs+)# **source-interface** *interface*	Assigns a source interface for a specific TACACS+ server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip tacacs source-interface** command.
Step 5	switch(config-tacacs+)# **exit**	Exits configuration mode.
Step 6	(Optional) switch(config)# **show tacacs-server groups**	Displays the TACACS+ server group configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure a TACACS+ server group:

```
switch# configure terminal
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# deadtime 30
switch(config-tacacs+)# exit
switch(config)# show tacacs-server groups
switch(config)# copy running-config startup-config
```

Configuring the Global Source Interface for TACACS+ Server Groups

You can configure a global source interface for TACACS+ server groups to use when accessing TACACS+ servers. You can also configure a different source interface for a specific TACACS+ server group.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip tacacs source-interface** *interface* **Example:** switch(config)# ip tacacs source-interface mgmt 0	Configures the global source interface for all TACACS+ server groups configured on the device. The source interface can be the management or the VLAN interface.
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits configuration mode.
Step 4	(Optional) **show tacacs-server** **Example:** switch# show tacacs-server	Displays the TACACS+ server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Specifying a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request by enabling the directed-request option. By default, a Cisco Nexus device forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as *username@hostname*, where *hostname* is the name of a configured RADIUS server.

✎

Note User specified logins are only supported for Telnet sessions.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server directed-request**	Displays the TACACS+ directed request configuration.

	Command or Action	**Purpose**
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring the Global TACACS+ Timeout Interval

You can set a global timeout interval that the Cisco Nexus device waits for responses from all TACACS+ servers before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from TACACS+ servers before declaring a timeout failure.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server timeout** *seconds*	Specifies the timeout interval for TACACS+ servers. The default timeout interval is 5 second and the range is from 1 to 60 seconds.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring the Timeout Interval for a Server

You can set a timeout interval that the Cisco Nexus device waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from a TACACS+ server before declaring a timeout failure.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **exit**	Exits configuration mode.
Step 3	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, the Cisco Nexus device uses port 49 for all TACACS+ requests.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **exit**	Exits configuration mode.
Step 3	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure TCP ports:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 port 2
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco Nexus device sends out a test packet.You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

The test idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco Nexus device sends out a test packet.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the number minutes before the Cisco Nexus device checks a TACACS+ server that was previously unresponsive. The default value

	Command or Action	Purpose
		is 0 minutes and the valid range is 0 to 1440 minutes.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure periodic TACACS+ server monitoring:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time
 3
switch(config)# tacacs-server dead-time 5
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco Nexus device waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

> **Note** When the dead-time interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server deadtime** *minutes*	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Manually Monitoring TACACS+ Servers or Groups

Procedure

	Command or Action	**Purpose**
Step 1	switch# **test aaa server tacacs**+ {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a TACACS+ server to confirm availability.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a TACACS+ server group to confirm availability.

Example

The following example shows how to manually issue a test message:

```
switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group TacGroup user2 As3He3CI
```

Disabling TACACS+

You can disable TACACS+.

⚠

Caution When you disable TACACS+, all related configurations are automatically discarded.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature tacacs**+	Disables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Displaying TACACS+ Statistics

To display the statistics, the switch maintains for TACACS+ activity, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}	Displays the TACACS+ statistics.

Example

For detailed information about the fields in the output from this command, see the *Command Reference* for your Nexus switch.

Verifying the TACACS+ Configuration

To display TACACS+ information, perform one of the following tasks:

Command	Purpose
show tacacs+ {**status** | **pending** | **pending-diff**}	Displays the TACACS+ Cisco Fabric Services distribution status and other details.
show running-config tacacs [**all**]	Displays the TACACS+ configuration in the running configuration.
show startup-config tacacs	Displays the TACACS+ configuration in the startup configuration.
show tacacs-serve [*host-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

Configuration Examples for TACACS+

This example shows how to configure TACACS+:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ToIkLhPpG"
switch(config)# tacacs-server host 10.10.2.2 key 7 "ShMoMhTl"
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# use-vrf management
```

This example shows how to enable tacacs+ and how to configure the tacacs+ server preshared keys to specify remote AAA servers to authenticate server group TacServer1:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ikvhw10"
switch(config)# tacacs-server host 1.1.1.1
switch(config)# tacacs-server host 1.1.1.2
```

```
switch(config)# aaa group server tacacs+ TacServer1
switch(config-tacacs+)#   server 1.1.1.1
switch(config-tacacs+)#   server 1.1.1.2
```

Default Settings for TACACS+

The following table lists the default settings for TACACS+ parameters.

Table 10: Default TACACS+ Parameters

Parameters	Default
TACACS+	Disabled
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on Cisco NX-OS devices and includes the following sections:

About LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP services are maintained in a database on an LDAP daemon running typically on a UNIX or Windows NT workstation. You must have access to and must configure an LDAP server before the configured LDAP features on your Cisco NX-OS device are available.

LDAP provides for separate authentication and authorization facilities. LDAP allows for a single access control server (the LDAP daemon) to provide each service authentication and authorization independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The LDAP client/server protocol uses TCP (port 389) for transport requirements. Cisco NX-OS devices provide centralized authentication using the LDAP protocol.

LDAP Authentication and Authorization

Clients establish a TCP connection and authentication session with an LDAP server through a simple bind (username and password). As part of the authorization process, the LDAP server searches its database to retrieve the user profile and other information.

You can configure the bind operation to first bind and then search, where authentication is performed first and authorization next, or to first search and then bind. The default method is to first search and then bind.

The advantage of searching first and binding later is that the distinguished name (DN) received in the search result can be used as the user DN during binding rather than forming a DN by prepending the username (cn attribute) with the baseDN. This method is especially helpful when the user DN is different from the username plus the baseDN. For the user bind, the bindDN is constructed as baseDN + append-with-baseDN, where append-with-baseDN has a default value of cn=$userid.

Note
As an alternative to the bind method, you can establish LDAP authentication using the compare method, which compares the attribute values of a user entry at the server. For example, the user password attribute can be compared for authentication. The default password attribute type is userPassword.

LDAP Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco NX-OS device using LDAP, the following actions occur:

1. When the Cisco NX-OS device establishes a connection, it contacts the LDAP daemon to obtain the username and password.

2. The Cisco NX-OS device eventually receives one of the following responses from the LDAP daemon:

   - ACCEPT—User authentication succeeds and service begins. If the Cisco NX-OS device requires user authorization, authorization begins.
   - REJECT—User authentication fails. The LDAP daemon either denies further access to the user or prompts the user to retry the login sequence.
   - ERROR—An error occurs at some time during authentication either at the daemon or in the network connection between the daemon and the Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS device tries to use an alternative method for authenticating the user.

   After authentication, the user also undergoes an additional authorization phase if authorization has been enabled on the Cisco NX-OS device. Users must first successfully complete LDAP authentication before proceeding to LDAP authorization.

3. If LDAP authorization is required, the Cisco NX-OS device again contacts the LDAP daemon, and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access. Services include the following:

   - Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
   - Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

Note
LDAP allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination but may include prompts for other items.

Note   In LDAP, authorization can occur before authentication.

LDAP Server Monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor an LDAP server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A Cisco NX-OS device periodically monitors dead LDAP servers and brings them to the alive state once they are responding. This process verifies that an LDAP server is in a working state before real AAA requests are sent its way. Whenever an LDAP server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated, and the Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance. The following figure shows the server states for LDAP server monitoring.

Figure 4: LDAP Server States
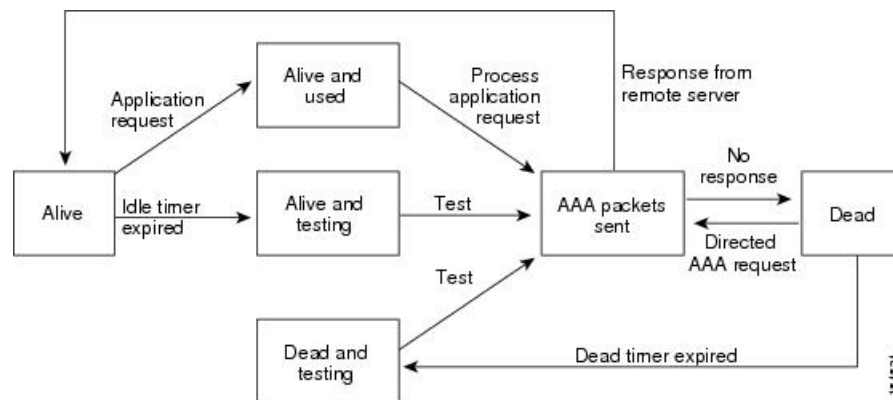
Note   The monitoring interval for alive servers and dead servers is different and can be configured by the user. The LDAP server monitoring is performed by sending a test authentication request to the LDAP server.

Vendor-Specific Attributes for LDAP

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the LDAP server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use.

Cisco VSA Format for LDAP

The Cisco LDAP implementation supports one vendor-specific option using the format recommended in the IETF specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is an = (equal sign) for mandatory attributes, and an * (asterisk) indicates optional attributes. When you use LDAP servers for authentication on a Cisco NX-OS device, LDAP directs the LDAP server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs. The following VSA protocol option is supported by the Cisco NX-OS software:

  • Shell—Protocol used in access-accept packets to provide user profile information.

The Cisco NX-OS software supports the following attribute:

  • roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space.

Virtualization Support for LDAP

The Cisco NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide.*

Licensing Requirements for LDAP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	LDAP requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for LDAP

LDAP has the following prerequisites:

  • Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
  • Ensure that the Cisco NX-OS device is configured as an LDAP client of the AAA servers.

Guidelines and Limitations for LDAP

LDAP has the following guidelines and limitations:

  • You can configure a maximum of 64 LDAP servers on the Cisco NX-OS device.

  • Cisco NX-OS supports only LDAP version 3.

  • Cisco NX-OS supports only these LDAP servers:

    • OpenLDAP

- Microsoft Active Directory

- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) version 1.

- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on a AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

Default Settings for LDAP

This table lists the default settings for LDAP parameters.

Parameters	Default
LDAP	Disabled
LDAP authentication method	First search and then bind
LDAP authentication mechanism	Plain
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	60 minutes
Periodic server monitoring username	test
Periodic server monitoring password	Cisco

Configuring LDAP

This section describes how to configure LDAP on a Cisco NX-OS device.

LDAP Server Configuration Process

You can configure LDAP servers by following this configuration process.

1. Enable LDAP.
2. Establish the LDAP server connections to the Cisco NX-OS device.
3. If needed, configure LDAP server groups with subsets of the LDAP servers for AAA authentication methods.
4. (Optional) Configure the TCP port.
5. (Optional) Configure the default AAA authorization method for the LDAP server.
6. (Optional) Configure an LDAP search map.
7. (Optional) If needed, configure periodic LDAP server monitoring.

Enabling or Disabling LDAP

By default, the LDAP feature is disabled on the Cisco NX-OS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **feature ldap** **Example:** `switch(config)# feature ldap`	Enables LDAP. Use the **no** form of this command to disable LDAP. **Note** When you disable LDAP, all related configurations are automatically discarded.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring LDAP Server Hosts

To access a remote LDAP server, you must configure the IP address or the hostname for the LDAP server on the Cisco NX-OS device. You can configure up to 64 LDAP servers.

Note By default, when you configure an LDAP server IP address or hostname on the Cisco NX-OS device, the LDAP server is added to the default LDAP server group. You can also add the LDAP server to another LDAP server group.

Before you begin

Enable LDAP.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

If you plan to enable the Secure Sockets Layer (SSL) protocol, make sure that the LDAP server certificate is manually configured on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} [**enable-ssl**]
	
	Example:
	
	`switch(config)# ldap-server host`
	`10.10.2.2 enable-ssl`
		Specifies the IPv4 or IPv6 address or hostname for an LDAP server.
		
		The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish an SSL session prior to sending the bind or search request.

Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring the RootDN for an LDAP Server

You can configure the root designated name (DN) for the LDAP server database. The rootDN is used to bind to the LDAP server to verify its state.

Before you begin

Enable LDAP.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] | **timeout** *seconds*]]	Specifies the rootDN for the LDAP server database and the bind password for the root. Optionally specifies the TCP port to use for LDAP messages to the server. The range is from

	Command or Action	Purpose
	Example: `switch(config)# ldap-server host 10.10.1.1 rootDN cn=manager,dc=acme,dc=com password Ur2Gd2BH timeout 60`	1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring LDAP Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must be configured to use LDAP. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time, but they take effect only when you apply them to an AAA service.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **aaa group server ldap** *group-name*
	
	Example:
	
	`switch(config)# aaa group server ldap LDAPServer1`
	`switch(config-ldap)#`
		Creates an LDAP server group and enters the LDAP server group configuration mode for that group.

Step 3	[**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:**	Configures the LDAP server as a member of the LDAP server group.

	Command or Action	**Purpose**
	switch(config-ldap)# server 10.10.2.2	If the specified LDAP server is not found, configure it using the **ldap-server host** command and retry this command.
Step 4	(Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]} **Example:** switch(config-ldap)# authentication compare password-attribute TyuL8r	Performs LDAP authentication using the bind or compare method. The default LDAP authentication method is the bind method using first search and then bind.
Step 5	(Optional) [**no**] **enable user-server-group** **Example:** switch(config-ldap)# enable user-server-group	Enables group validation. The group name should be configured in the LDAP server. Users can login through public-key authentication only if the username is listed as a member of this configured group in the LDAP server.
Step 6	(Optional) [**no**] **enable Cert-DN-match** **Example:** switch(config-ldap)# enable Cert-DN-match	Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login.
Step 7	(Optional) [**no**] **use-vrf** *vrf-name* **Example:** switch(config-ldap)# use-vrf vrf1	Specifies the VRF to use to contact the servers in the server group.
Step 8	**exit** **Example:** switch(config-ldap)# exit switch(config)#	Exits LDAP server group configuration mode.
Step 9	(Optional) **show ldap-server groups** **Example:** switch(config)# show ldap-server groups	Displays the LDAP server group configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring the Global LDAP Timeout Interval

You can set a global timeout interval that determines how long the Cisco NX-OS device waits for responses from all LDAP servers before declaring a timeout failure.

Before you begin

Enable LDAP.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ldap-server timeout** *seconds* **Example:** switch(config)# ldap-server timeout 10	Specifies the timeout interval for LDAP servers. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds.
Step 3	(Optional) **show ldap-server** **Example:** switch(config)# show ldap-server	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring the Timeout Interval for an LDAP Server

You can set a timeout interval that determines how long the Cisco NX-OS device waits for responses from an LDAP server before declaring a timeout failure.

Before you begin

Enable LDAP.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* **Example:**	Specifies the timeout interval for a specific server. The default is the global value.

	Command or Action	Purpose
	`switch(config)# ldap-server host server1 timeout 10`	**Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring TCP Ports

You can configure another TCP port for the LDAP servers if there are conflicts with another application. By default, Cisco NX-OS devices use port 389 for all LDAP requests.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port* [**timeout** *seconds*] **Example:** `switch(config)# ldap-server host 10.10.1.1 port 200 timeout 5`	Specifies the TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535. Optionally specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. **Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.
Step 3	(Optional) **show ldap-server** **Example:**	Displays the LDAP server configuration.

	Command or Action	Purpose
	switch(config)# show ldap-server	
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring LDAP Search Maps

You can configure LDAP search maps to send a search query to the LDAP server. The server searches its database for data meeting the criteria specified in the search map.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**ldap search-map** *map-name* **Example:** switch(config)# ldap search-map map1 switch(config-ldap-search-map)#	Configures an LDAP search map.
Step 3	(Optional) [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name* **Example:** switch(config-ldap-search-map)# userprofile attribute-name att-name search-filter (&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com	Configures the attribute name, search filter, and base-DN for the user profile, trusted certificate, CRL, certificate DN match, public key match, or user-switchgroup lookup search operation. These values are used to send a search query to the LDAP server. The *attribute-name* argument is the name of the attribute in the LDAP server that contains the Nexus role definition.
Step 4	(Optional) **exit** **Example:** switch(config-ldap-search-map)# exit switch(config)#	Exits LDAP search map configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show ldap-search-map** **Example:** `switch(config)# show ldap-search-map`	Displays the configured LDAP search maps.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Periodic LDAP Server Monitoring

You can monitor the availability of LDAP servers. The configuration parameters include the username and password to use for the server, the rootDN to bind to the server to verify its state, and an idle timer. The idle timer specifies the interval in which an LDAP server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the LDAP database.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]] **Example:** `switch(config)# ldap-server host 10.10.1.1 test rootDN root1 username user1 password Ur2Gd2BH idle-time 3`	Specifies the parameters for server monitoring. The default username is test, and the default password is Cisco. The default value for the idle timer is 60 minutes, and the valid range is from 1 to 1440 minutes. **Note** We recommend that the user not be an existing user in the LDAP server database.
Step 3	[**no**] **ldap-server deadtime** *minutes* **Example:**	Specifies the number of minutes before the Cisco NX-OS device checks an LDAP server

	Command or Action	Purpose
	switch(config)# ldap-server deadtime 5	that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 60 minutes.
Step 4	(Optional) **show ldap-server** **Example:** switch(config)# show ldap-server	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring the LDAP Dead-Time Interval

You can configure the dead-time interval for all LDAP servers. The dead-time interval specifies the time that the Cisco NX-OS device waits, after declaring that an LDAP server is dead, before sending out a test packet to determine if the server is now alive.

> **Note** When the dead-time interval is 0 minutes, LDAP servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ldap-server deadtime** *minutes* **Example:** switch(config)# ldap-server deadtime 5	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 60 minutes.
Step 3	(Optional) **show ldap-server** **Example:** switch(config)# show ldap-server	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

Command or Action	Purpose
| ```
switch(config)# copy running-config
startup-config
``` | |

Configuring AAA Authorization on LDAP Servers

You can configure the default AAA authorization method for LDAP servers.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose
| **Step 1** | **configure terminal**<br><br>**Example:**<br>```
switch# configure terminal
switch(config)#
``` | Enters global configuration mode. |
| **Step 2** | **aaa authorization** {**ssh-certificate** \| **ssh-publickey**} **default** {**group** *group-list* \| **local**}<br><br>**Example:**<br>```
switch(config)# aaa authorization
ssh-certificate default group LDAPServer1
 LDAPServer2
``` | Configures the default AAA authorization method for the LDAP servers.<br><br>The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the **ssh-publickey** keyword configures LDAP or local authorization with the SSH public key. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.<br><br>The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The **local** method uses the local database for authorization. |
| **Step 3** | (Optional) **show aaa authorization** [**all**]<br><br>**Example:**<br>```
switch(config)# show aaa authorization
``` | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>```
switch(config)# copy running-config
startup-config
``` | Copies the running configuration to the startup configuration. |

Monitoring LDAP Servers

You can monitor the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
	
	Example:
	`switch# show ldap-server statistics`
	`10.10.1.1`
		Displays the LDAP server statistics.

Clearing LDAP Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	(Optional) **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
	
	Example:
	`switch# show ldap-server statistics`
	`10.10.1.1`
		Displays the LDAP server statistics.

Step 2	**clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
	
	Example:
	`switch# clear ldap-server statistics`
	`10.10.1.1`
		Clears the LDAP server statistics.

Verifying the LDAP Configuration

To display LDAP configuration information, perform one of the following tasks.

Command	Purpose
show running-config ldap [**all**]	Displays the LDAP configuration in the running configuration.
show startup-config ldap	Displays the LDAP configuration in the startup configuration.
show ldap-server	Displays LDAP configuration information.
show ldap-server groups	Displays LDAP server group configuration information.
show ldap-server statistics {*hostname* | *ipv4-address* | *ipv6-address*}	Displays LDAP statistics.
show ldap-search-map	Displays information about the configured LDAP attribute maps.

Configuration Examples for LDAP

The following example shows how to configure an LDAP server host and server group:

```
feature ldap
ldap-server host 10.10.2.2 enable-ssl
aaa group server ldap LdapServer
server 10.10.2.2
exit
show ldap-server
show ldap-server groups
```

The following example shows how to configure an LDAP search map:

```
ldap search-map s0
userprofile attribute-name att-name search-filter "
(&(objectClass=Person)(sAMAccountName=$userid))" base-DN dc=acme,dc=com
exit
show ldap-search-map
```

The following example shows how to configure AAA authorization with certificate authentication for an LDAP server:

```
aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
exit
show aaa authorization
```

The following example shows how you can validate the authentication:

```
failing
test aaa group LdapServer user <user-password>
user has failed authentication
```

```
! working
test aaa group LdapServer user <user-password>
user has been authenticated
```

Where to Go Next

You can now configure AAA authentication methods to include the server groups.

Additional References for LDAP

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Configuring SSH and Telnet

This chapter contains the following sections:

Information About SSH and Telnet

SSH Server

The Secure Shell Protocol (SSH) server feature enables a SSH client to make a secure, encrypted connection to a Cisco Nexus device. SSH uses strong encryption for authentication. The SSH server in the Cisco Nexus device switch interoperates with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, and the use of locally stored user names and passwords.

SSH Client

The SSH client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a switch to make a secure, encrypted connection to another Cisco Nexus device or to any other device running an SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco Nexus device works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco NX-OS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)
- SSH version 2 using the Elliptic Curve Digital Signature Algorithm (ECDSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts the following types of key-pairs for use by SSH version 2:

- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol.
- The **ecdsa** option generates the ECDSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)

⚠️

Caution   If you delete all of the SSH keys, you cannot start the SSH services.

SSH Authentication Using Digital Certificates

SSH authentication on Cisco NX-OS devices provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs.

You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.

You can configure SSH authentication using X.509v3 certificates (RFC 6187). X.509v3 certificate-based SSH authentication uses certificates combined with a smartcard to enable two-factor authentication for Cisco device access. The SSH client is provided by Cisco partner Pragma Systems.

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site, and then passes the keystrokes from one system to the other. Telnet can accept either an IP address or a domain name as the remote system address.

The Telnet server is enabled by default on the Cisco Nexus device.

Guidelines and Limitations for SSH

SSH has the following configuration guidelines and limitations:

- The Cisco Nexus device supports only SSH version 2 (SSHv2).

- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.

Configuring SSH

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**ssh key** {**dsa** [**force**] | **rsa** [*bits*[**force**]] | **ecdsa** [*bits* [**force**]]}	Generates the SSH server key.
		The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048. The default value is 1024.
		You cannot specify the size of the DSA key. It is always set to 1024 bits.
		Use the **force** keyword to replace an existing key.
		Note If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all
Step 3	**ssh rekey max-data** *max-data* **max-time** *max-time*i	Configures the rekey parameters.

Command or Action	Purpose
Example: `switch(config)# ssh rekey max-data 1K max-time 1M`	
Step 4 **feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.
Step 5 switch(config)# **exit**	Exits global configuration mode.
Step 6 (Optional) **show ssh key** [**dsa** | **rsa** | **ecdsa**] [**md5**] **Example:** `switch# show ssh key`	Displays the SSH server keys. This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
Step 7 **show run security all**	
Step 8 (Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to generate an SSH server key:

```
switch# configure terminal
switch(config)# ssh key rsa 2048
switch(config)# exit
switch# show ssh key
switch# copy running-config startup-config
```

Specifying the SSH Public Keys for User Accounts

You can configure an SSH public key to log in using an SSH client without being prompted for a password. You can specify the SSH public key in one of three different formats:

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Specifying the SSH Public Keys in Open SSH Format

You can specify the SSH public keys in SSH format for user accounts.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in SSH format.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify an SSH public key in open SSH format:

```
switch# configure terminal
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Note   The **username** command in the example above is a single line that has been broken for legibility.

Specifying the SSH Public Keys in IETF SECSH Format

You can specify the SSH public keys in IETF SECSH format for user accounts.

Procedure

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in IETF SECSH format from a server. The server can be FTP, SCP, SFTP, or TFTP.
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **username** *username* **sshkey file** *filename*	Configures the SSH public key in SSH format.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show user-account**	Displays the user account configuration.

	Command or Action	Purpose
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public key in the IETF SECSH format:

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Specifying the SSH Public Keys in PEM-Formatted Public Key Certificate Form

You can specify the SSH public keys in PEM-formatted Public Key Certificate form for user accounts.

Procedure

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public keys in PEM-formatted public key certificate form:

```
switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem
switch# configure terminal
switch# show user-account
switch# copy running-config startup-config
```

Configuring the SSH Source Interface

You can configure SSH to use a specific interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip ssh source-interface** *type slot/port*	Configures the source interface for all SSH packets. The following list contains the valid values for *interface*. • ethernet • loopback • mgmt • port-channel • vlan
Step 3	switch(config)# **show ip ssh source-interface**	Displays the configured SSH source interface.

Example

This example shows how to configure the SSH source interface:

```
switch(config)# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip ssh source-interface ethernet 1/7
switch(config)# show ip ssh source-interface
VRF Name                                Interface
default                                 Ethernet1/7
```

Starting SSH Sessions to Remote Devices

You can start SSH sessions to connect to remote devices from your Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **ssh** {*hostname*	*username@hostname*} [**vrf** *vrf-name*]	Creates an SSH session to a remote device. The *hostname* argument can be an IPv4 address or a hostname.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, you establish a trusted SSH relationship with that server.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear ssh hosts**	Clears the SSH host sessions.

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature ssh**	Enables/disables the SSH server. The default is enabled.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show ssh server**	Displays the SSH server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys after you disable the SSH server.

Note   To reenable SSH, you must first generate an SSH server key.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server.
Step 3	switch(config)# **no ssh key** [**dsa** | **rsa**]	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show ssh key**	Displays the SSH server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user SSH session.

Configuration Examples for SSH

The following example shows how to configure SSH:

Procedure

Step 1   Generate an SSH server key.

```
switch(config)# ssh key rsa
generating rsa key(1024 bits).....
.
generated rsa key
```

Step 2   Enable the SSH server.

```
switch# configure terminal
switch(config)# feature ssh
```

Note   This step should not be required because the SSH server is enabled by default.

Step 3   Display the SSH server key.

```
switch(config)# show ssh key
rsa Keys generated:Fri May  8 22:09:47 2009

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYzCfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZ/
cTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4ZXIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5/
Ninn0Mc=

bitcount:1024
fingerprint:
4b:4d:f6:b9:42:e9:d9:71:3c:bd:09:94:4a:93:ac:ca
**************************************
could not retrieve dsa key information
**************************************
```

Step 4   Specify the SSH public key in Open SSH format.

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz

CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z

XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
```

Step 5 Save the configuration.

```
switch(config)# copy running-config startup-config
```

Configuring X.509v3 Certificate-Based SSH Authentication

You can configure SSH authentication using X.509v3 certificates.

Before you begin

Enable the SSH server on the remote device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	**username** *user-id* [**password** [**0** | **5**] *password*]
	
	Example:
	```switch(config)# username jsmith password```
	``` 4Ty18Rnt```
		Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.
		
		Usernames must begin with an alphanumeric character.
		
		The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text).
		
		Note If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device.

		Command or Action	Purpose
		Note	If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.
Step 3		**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**} **Example:** `switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`	Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.
Step 4		[**no**] **crypto ca trustpoint** *trustpoint* **Example:** `switch(config)# crypto ca trustpoint winca`	Configures a trustpoint.
Step 5		[**no**] **crypto ca authentication** *trustpoint* **Example:** `switch(config)# crypto ca authentication winca`	Configures a certificate chain for the trustpoint.
Step 6		**crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl** **Example:** `switch(config)# crypto ca crl request winca bootflash:crllist.crl`	Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA). Note Static CRL is the only supported revocation check method.
Step 7		(Optional) **show crypto ca certificates** **Example:** `switch(config)# show crypto ca certificates`	Displays the configured certificate chain and associated trustpoint.
Step 8		(Optional) **show crypto ca crl** *trustpoint* **Example:** `switch(config)# show crypto ca crl winca`	Displays the contents of the CRL list of the specified trustpoint.
Step 9		(Optional) **show user-account** **Example:** `switch(config)# show user-account`	Displays configured user account details.

	Command or Action	Purpose
Step 10	(Optional) **show users** **Example:** `switch(config)# show users`	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Example for X.509v3 Certificate-Based SSH Authentication

The following example shows how to configure SSH authentication using X.509v3 certificates:

```
configure terminal
username jsmith password 4Ty18Rnt
username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
rsa
crypto ca trustpoint tp1
crypto ca authentication tp1
crypto ca crl request tp1 bootflash:crl1.crl

show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
subject= /CN=SecDevCA
issuer= /CN=SecDevCA
serial=01AB02CD03EF04GH05IJ06KL07MN
notBefore=Jun 29 12:36:26 2016 GMT
notAfter=Jun 29 12:46:23 2021 GMT
SHA1 Fingerprint=47:29:E3:00:C1:C1:47:F2:56:8B:AC:B2:1C:64:48:FC:F4:8D:53:AF
purposes: sslserver sslclient

show crypto ca crl tp1
Trustpoint: tp1 CRL: Certificate Revocation List (CRL):
    Version 2 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: /CN=SecDevCA
    Last Update: Aug 8 20:03:15 2016 GMT
    Next Update: Aug 16 08:23:15 2016 GMT
    CRL extensions:
        X509v3 Authority Key Identifier:
            keyid:30:43:AA:80:10:FE:72:00:DE:2F:A2:17:E4:61:61:44:CE:78:FF:2A

show user-account
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh cert DN : /C = US, ST = New York, L = Metropolis, O = cisco , OU = csg, CN =
user1; Algo: x509v3-sign-rsa

show users
```

```
NAME       LINE       TIME          IDLE     PID        COMMENT
user1      pts/1      Jul 27 18:43  00:03    18796      (10.10.10.1)   session=ssh
```

Configuring Legacy SSH Algorithm Support

You can configure support for legacy SSH security algorithms, message authentication codes (MACs), key types, and ciphers.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **ssh kexalgos all** **Example:** `switch(config)# ssh kexalgos all`	Enables all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys. Supported KexAlgorithmns are: • curve25519-sha256 • diffie-hellman-group-exchange-sha256 • diffie-hellman-group14-sha1 • diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521
Step 3	(Optional) **ssh macs all** **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512 • aes256-gcm@openssh.com • aes128-gcm@openssh.com

	Command or Action	Purpose

Step 4	(Optional) **ssh ciphers all**
	
	Example:
	
	`switch(config)# ssh ciphers all`
		Enables all supported ciphers to encrypt the connection.
		
		Supported ciphers are:
		
		   • aes128-cbc
		
		   • aes192-cbc
		
		   • aes256-cbc
		
		   • aes128-ctr
		
		   • aes192-ctr
		
		   • aes256-ctr
		
		   • aes256-gcm@openssh.com
		
		   • aes128-gcm@openssh.com

Step 5	(Optional) **ssh keytypes all**
	
	Example:
	
	`switch(config)# ssh keytypes all`
		Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client.
		
		Supported key types are:
		
		   • ecdsa-sha2-nistp256
		
		   • ecdsa-sha2-nistp384
		
		   • ecdsa-sha2-nistp521
		
		   • ssh-dss
		
		   • ssh-rsa

Changing the Default SSH Server Port

Beginning with Cisco NX-OS Cisco Release 9.2(1), you can change the SSHv2 port number from the default port number 22. Encryptions used while changing the default SSH port provides you with connections that support stronger privacy and session integrity

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**show sockets** *local-port-range*
	
	Example:
	
	`switch(config)# show sockets local port`
	`range (15001 - 58000)`
	`switch(config)# local port range (58001`
	`- 63535) and nat port range (63536 -`
	`65535)`
	
	`switch# show sockets local-port-range`
	`Kstack local port range (15001 - 22002)`
	`Netstack local port range (22003 - 65535)`
		Displays the available port range.

Step 4	**ssh port** *local-port* **Example:** `switch(config)# ssh port 58003`	Configures the port. **Note** When you upgrade from an earlier release to Release 9.3(1) or later releases, ensure that features with user-defined SSH port, are within the following range: • For Release 9.3(1) and Release 9.3(2): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 - 63535, and nat port range is from 63536 to 65535 • From Release 9.3(3): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 to 60535, and nat port range is from 60536 to 65535
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

Step 6	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 7	(Optional) **show running-config security all** **Example:**	Displays the security configuration.

	Command or Action	Purpose
	switch# ssh port 58003	
Step 8	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring Telnet

Enabling the Telnet Server

By default, the Telnet server is enabled. You can disable the Telnet server on your Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature telnet**	Enables/disables the Telnet server. The default is enabled.

Reenabling the Telnet Server

If the Telnet server on your Cisco Nexus device has been disabled, you can reenable it.

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **[no] feature telnet**	Reenables the Telnet server.

Configuring the Telnet Source Interface

You can configure Telnet to use a specific interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip telnet source-interface** *type slot/port*	Configures the source interface for all Telnet packets. The following list contains the valid values for *interface*. • ethernet

	Command or Action	Purpose
		• loopback
		• mgmt
		• port-channel
		• vlan

Example

This example shows how to configure the Telnet source interface:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip telnet source-interface ethernet 1/6
switch(config)# show ip telnet source-interface
VRF Name                                 Interface
default                                  Ethernet1/6
switch(config)#
```

Starting Telnet Sessions to Remote Devices

Before you start a Telnet session to connect to remote devices, you should do the following:

• Obtain the hostname for the remote device and, if needed, obtain the username on the remote device.

• Enable the Telnet server on the Cisco Nexus device.

• Enable the Telnet server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet** *hostname*	Creates a Telnet session to a remote device. The *hostname* argument can be an IPv4 address, an IPv6 address, or a device name.

Example

The following example shows how to start a Telnet session to connect to a remote device:

```
switch# telnet 10.10.1.1
Trying 10.10.1.1...
Connected to 10.10.1.1.
Escape character is '^]'.
switch login:
```

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

To display the SSH configuration information, perform one of the following tasks:

Command or Action	Purpose
switch# **show ssh key [dsa	rsa][md5]**	Displays SSH server keys. This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
switch# **show running-config security [all]**	Displays the SSH and user account configuration in the running configuration. The all keyword displays the default values for the SSH and user accounts.
switch# **show ssh server**	Displays the SSH server configuration.
switch# **show user-account**	Displays user account information.
switch# **show users**	Displays the users logged into the device.
switch# **show crypto ca certificates**	Displays the configured certificate chain and associated trustpoint for X.509v3 certificate-based SSH authentication.
switch# **show crypto ca crl** *trustpoint*	Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication.

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 11: Default SSH Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits

Parameters	Default
RSA key bits for generation	1024
Telnet server	Disabled

C H A P T E R **9**

Configuring PKI

This chapter contains the following sections:

Information About PKI

This section provides information about PKI.

CAs and Digital Certificates

Certificate authorities (CAs) manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each device or user has a key pair that contains both a private key and a public key. The private key is kept secret and is known only to the owning device or user only. However, the public key is known to everybody. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates link the digital signature to the sender. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The CA that signs the certificate is a third party that the receiver explicitly trusts to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Typically, this process is handled out of band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default.

Trust Model, Trust Points, and Identity CAs

The PKI trust model is hierarchical with multiple configurable trusted CAs. You can configure each participating device with a list of trusted CAs so that a peer certificate obtained during the security protocol exchanges can be authenticated if it was issued by one of the locally trusted CAs. The Cisco NX-OS software locally stores the self-signed root certificate of the trusted CA (or certificate chain for a subordinate CA). The process of securely obtaining a trusted CA's root certificate (or the entire chain in the case of a subordinate CA) and storing it locally is called *CA authentication*.

The information about a trusted CA that you have configured is called the *trust point* and the CA itself is called a *trust point CA*. This information consists of a CA certificate (or certificate chain in case of a subordinate CA) and certificate revocation checking information.

The Cisco NX-OS device can also enroll with a trust point to obtain an identity certificate to associate with a key pair. This trust point is called an *identity CA*.

RSA Key Pairs and Identity Certificates

You can obtain an identity certificate by generating one or more RSA key pairs and associating each RSA key pair with a trust point CA where the Cisco NX-OS device intends to enroll. The Cisco NX-OS device needs only one identity per CA, which consists of one key pair and one identity certificate per CA.

The Cisco NX-OS software allows you to generate RSA key pairs with a configurable key size (or modulus). The default key size is 512. You can also configure an RSA key-pair label. The default key label is the device fully qualified domain name (FQDN).

The following list summarizes the relationship between trust points, RSA key pairs, and identity certificates:

- A trust point corresponds to a specific CA that the Cisco NX-OS device trusts for peer certificate verification for any application (such as SSH).

- A Cisco NX-OS device can have many trust points and all applications on the device can trust a peer certificate issued by any of the trust point CAs.

- A trust point is not restricted to a specific application.

- A Cisco NX-OS device enrolls with the CA that corresponds to the trust point to obtain an identity certificate. You can enroll your device with multiple trust points which means that you can obtain a separate identity certificate from each trust point. The identity certificates are used by applications depending upon the purposes specified in the certificate by the issuing CA. The purpose of a certificate is stored in the certificate as a certificate extension.

- When enrolling with a trust point, you must specify an RSA key pair to be certified. This key pair must be generated and associated to the trust point before generating the enrollment request. The association between the trust point, key pair, and identity certificate is valid until it is explicitly removed by deleting the certificate, key pair, or trust point.

- The subject name in the identity certificate is the fully qualified domain name for the Cisco NX-OS device.

- You can generate one or more RSA key pairs on a device and each can be associated to one or more trust points. But no more than one key pair can be associated to a trust point, which means only one identity certificate is allowed from a CA.

- If the Cisco NX-OS device obtains multiple identity certificates (each from a distinct CA), the certificate that an application selects to use in a security protocol exchange with a peer is application specific.

- You do not need to designate one or more trust points for an application. Any application can use any certificate issued by any trust point as long as the certificate purpose satisfies the application requirements.

- You do not need more than one identity certificate from a trust point or more than one key pair to be associated to a trust point. A CA certifies a given identity (or name) only once and does not issue multiple certificates with the same name. If you need more than one identity certificate for a CA and if the CA allows multiple certificates with the same names, you must define another trust point for the same CA, associate another key pair to it, and have it certified.

Multiple Trusted CA Support

The Cisco NX-OS device can trust multiple CAs by configuring multiple trust points and associating each with a distinct CA. With multiple trusted CAs, you do not have to enroll a device with the specific CA that issued the certificate to a peer. Instead, you can configure the device with multiple trusted CAs that the peer trusts. The Cisco NX-OS device can then use a configured trusted CA to verify certificates received from a peer that were not issued by the same CA defined in the identity of the peer device.

PKI Enrollment Support

Enrollment is the process of obtaining an identity certificate for the device that is used for applications like SSH. It occurs between the device that requests the certificate and the certificate authority.

The Cisco NX-OS device performs the following steps when performing the PKI enrollment process:

- Generates an RSA private and public key pair on the device.

- Generates a certificate request in standard format and forwards it to the CA.

Note The CA administrator may be required to manually approve the enrollment request at the CA server, when the request is received by the CA.

- Receives the issued certificate back from the CA, signed with the CA's private key.

- Writes the certificate into a nonvolatile storage area on the device (bootflash).

Manual Enrollment Using Cut-and-Paste

The Cisco NX-OS software supports certificate retrieval and enrollment using manual cut-and-paste. Cut-and-paste enrollment means that you must cut and paste the certificate requests and resulting certificates between the device and the CA.

You must perform the following steps when using cut and paste in the manual enrollment process:

- Create an enrollment certificate request, which the Cisco NX-OS device displays in base64-encoded text form.

- Cut and paste the encoded certificate request text in an e-mail or in a web form and send it to the CA.

- Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail or in a web browser download.

• Cut and paste the issued certificate to the device using the certificate import facility.

Multiple RSA Key Pair and Identity CA Support

Multiple identity CAs enable the device to enroll with more than one trust point, which results in multiple identity certificates, each from a distinct CA. With this feature, the Cisco NX-OS device can participate in SSH and other applications with many peers using certificates issued by CAs that are acceptable to those peers.

The multiple RSA key-pair feature allows the device to maintain a distinct key pair for each CA with which it is enrolled. It can match policy requirements for each CA without conflicting with the requirements specified by the other CAs, such as the key length. The device can generate multiple RSA key pairs and associate each key pair with a distinct trust point. Thereafter, when enrolling with a trust point, the associated key pair is used to construct the certificate request.

Peer Certificate Verification

The PKI support on a Cisco NX-OS device can verify peer certificates. The Cisco NX-OS software verifies certificates received from peers during security exchanges for applications, such as SSH. The applications verify the validity of the peer certificates. The Cisco NX-OS software performs the following steps when verifying peer certificates:

- Verifies that the peer certificate is issued by one of the locally trusted CAs.

- Verifies that the peer certificate is valid (not expired) with respect to current time.

- Verifies that the peer certificate is not yet revoked by the issuing CA.

For revocation checking, the Cisco NX-OS software supports the certificate revocation list (CRL). A trust point CA can use this method to verify that the peer certificate has not been revoked.

Certificate Revocation Checking

The Cisco NX-OS software can check the revocation status of CA certificates. The applications can use the revocation checking mechanisms in the order that you specify. The choices are CRL,OCSP, none, or a combination of these methods.

CRL Support

The CAs maintain certificate revocation lists (CRLs) to provide information about certificates revoked prior to their expiration dates. The CAs publish the CRLs in a repository and provide the download public URL in all issued certificates. A client verifying a peer's certificate can obtain the latest CRL from the issuing CA and use it to determine if the certificate has been revoked. A client can cache the CRLs of some or all of its trusted CAs locally and use them later if necessary until the CRLs expire.

The Cisco NX-OS software allows the manual configuration of predownloaded CRLs for the trust points, and then caches them in the device bootflash (cert-store). During the verification of a peer certificate, the Cisco NX-OS software checks the CRL from the issuing CA only if the CRL has already been cached locally and the revocation checking is configured to use the CRL. Otherwise, the Cisco NX-OS software does not perform CRL checking and considers the certificate to be not revoked unless you have configured other revocation checking methods.

NDcPP: OCSP for Syslog

Online Certificate Status Protocol (OCSP) is a method to check certificate revocation when a peer has to retrieve this revocation information and then validate it to check the certificate revocation status. In this method, the certification revocation status is limited by the peer's ability to reach an OCSP responder through the cloud or by the certificate sender's performance in retrieving the certificate revocation-information.

When the remote syslog server shares the certificate which has an OCSP responder URL, the client sends the server certificate to an external OCSP responder (CA) server. The CA server validates this certificate and confirms if it is a valid or a revoked certificate. In this case, the client does not have to maintain the revoked certificate list locally.

Import and Export Support for Certificates and Associated Key Pairs

As part of the CA authentication and enrollment process, the subordinate CA certificate (or certificate chain) and identity certificates can be imported in standard PEM (base64) format.

The complete identity information in a trust point can be exported to a file in the password-protected PKCS#12 standard format. It can be later imported to the same device (for example, after a system crash) or to a replacement device. The information in a PKCS#12 file consists of the RSA key pair, the identity certificate, and the CA certificate (or chain).

Licensing Requirements for PKI

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	The PKI feature requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for PKI

PKI has the following configuration guidelines and limitations:

- The maximum number of key pairs you can configure on a Cisco NX-OS device is 16.

- The maximum number of trust points you can declare on a Cisco NX-OS device is 16.

- The maximum number of identify certificates that you can configure on a Cisco NX-OS device is 16.

- The maximum number of certificates in a CA certificate chain is 10.

- The maximum number of trust points you can authenticate to a specific CA is 10.

- Configuration rollbacks do not support the PKI configuration.

- Beginning with Cisco NX-OS Release 9.3(5), Cisco NX-OS software supports NDcPP: OCSP for Syslog.

Note	If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for PKI

This table lists the default settings for PKI parameters.

Table 12: Default PKI Parameters

Parameters	Default
Trust point	None
RSA key pair	None
RSA key-pair label	Device FQDN
RSA key-pair modulus	512
RSA key-pair exportable	Enabled
Revocation check method	CRL

Configuring CAs and Digital Certificates

This section describes the tasks that you must perform to allow CAs and digital certificates on your Cisco NX-OS device to interoperate.

Configuring the Hostname and IP Domain Name

You must configure the hostname and IP domain name of the device if you have not yet configured them because the Cisco NX-OS software uses the fully qualified domain name (FQDN) of the device as the subject in the identity certificate. Also, the Cisco NX-OS software uses the device FQDN as a default key label when you do not specify a label during key-pair generation. For example, a certificate named DeviceA.example.com is based on a device hostname of DeviceA and a device IP domain name of example.com.

Caution	Changing the hostname or IP domain name after generating the certificate can invalidate the certificate.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hostname** *hostname* **Example:** `switch(config)# hostname DeviceA`	Configures the hostname of the device.

Step 3	**ip domain-name** *name* [**use-vrf** *vrf-name*]
	
	Example:
	
	`DeviceA(config)# ip domain-name`
	`example.com`
		Configures the IP domain name of the device. If you do not specify a VRF name, the command uses the default VRF.

Step 4	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show hosts** **Example:** `switch# show hosts`	Displays the IP domain name.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Generating an RSA Key Pair

You can generate an RSA key pairs to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the RSA key pair before you can obtain a certificate for your device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]	Generates an RSA key pair. The maximum number of key pairs on a device is 16.

	Command or Action	Purpose
	Example: `switch(config)# crypto key generate rsa exportable`	The label string is alphanumeric, case sensitive, and has a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.).
		Valid modulus values are 512, 768, 1024, 1536, and 2048. The default modulus size is 512.
		Note The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus.
		By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format.
		Caution You cannot change the exportability of a key pair.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** `switch# show crypto key mypubkey rsa`	Displays the generated key.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a Trust Point CA Association

You must associate the Cisco NX-OS device with a trust point CA.

Before you begin

Generate the RSA key pair.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	``switch# configure terminal
switch(config)#```	
Step 2	**crypto ca trustpoint** *name* **Example:** ```switch(config)# crypto ca trustpoint
admin-ca
switch(config-trustpoint)#```	Declares a trust point CA that the device should trust and enters trust point configuration mode. **Note** The maximum number of trust points that you can configure on a device is 16.
Step 3	**enrollment terminal** **Example:** ```switch(config-trustpoint)# enrollment
terminal```	Enables manual cut-and-paste certificate enrollment. The default is enabled. **Note** The Cisco NX-OS software supports only the manual cut-and-paste method for certificate enrollment.
Step 4	**rsakeypair** *label* **Example:** ```switch(config-trustpoint)# rsakeypair
SwitchA```	Specifies the label of the RSA key pair to associate to this trust point for enrollment. **Note** You can specify only one RSA key pair per CA.
Step 5	**exit** **Example:** ```switch(config-trustpoint)# exit
switch(config)#```	Exits trust point configuration mode.
Step 6	(Optional) **show crypto ca trustpoints** **Example:** ```switch(config)# show crypto ca
trustpoints```	Displays trust point information.
Step 7	(Optional) **copy running-config startup-config** **Example:** ```switch(config)# copy running-config
startup-config```	Copies the running configuration to the startup configuration.

Authenticating the CA

The configuration process of trusting a CA is complete only when the CA is authenticated to the Cisco NX-OS device. You must authenticate your Cisco NX-OS device to the CA by obtaining the self-signed certificate of the CA in PEM format, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

> ✎
>
> **Note**   The CA that you are authenticating is not a self-signed CA when it is a subordinate CA to another CA, which itself may be a subordinate to yet another CA, and so on, finally ending in a self-signed CA. This type of CA certificate is called the *CA certificate chain* of the CA being authenticated. In this case, you must input the full list of the CA certificates of all the CAs in the certification chain during the CA authentication. The maximum number of certificates in a CA certificate chain is 10.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca authenticate** *name*
	
	Example:
	`switch(config)# crypto ca authenticate`
	`admin-ca`
	`input (cut & paste) CA certificate`
	`(chain) in PEM format;`
	`end the input with a line containing only`
	` END OF INPUT :`
	`-----BEGIN CERTIFICATE-----`
	`MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRl1jK0ZejANBgkqhkiG9w0BAQUFADCB`
	`kDEgMB4GCSqGSIb3DQEJARYRYWlhbmRRzZLBjaXNjby5jb20xCzAJBgNVBAYTAklO`
	`MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTIEOMAwGA1UE`
	`ChMFQ2lzY28xEzARBgNVBAsTCHf5ldHNvb3JhZZUkEjAQBgNVBAMTCUFwYXJuYSBD`
	`QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIIGQMSAwHgYJKoZIhvcN`
	`AQkBFhFhbWFuZZGtlQGNpc2NvLmNvbXvbtEIMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth`
	`cnbhdGFrYTIESMBAGA1UEBxMQnFuZ2Fsb3JlIMQ4wDAYDVQQKEwVDaXNjbzEIMBEG`
	`A1UECxMKbnV0c3RvcmFtZIESMBAGA1UEAxMQQXBhcmFhIENBMFwwDQYJKoZIhvcN`
	`AQEBBQADSwAwSAIBAMW/7b3+DKJPANBsIHHzluNcdM87ypyzwuoSNZXOMpeRXXI`
	`OzyBAgriXIZASFUUOwQ1iIM8rO/41jf8RxvVKvysCAwEAAaOBvzCBvDALBgNVHQ8E`
	`BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRnQ`
	`GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs`
	`L0FwYXJuYSUyMENBLmNybaAwoO6gLIYqbmlsZIovLlxxc3NlL1TA4XENlcrR-EcnJv`
	`bGwcQXBhcnbhJTIwQEIiY3JsMBAGCSsGAQQBgcjcVAQQDAgEAMA0GCSqGSIb3DQEB`
	`BQUAA0EAHv6UQ+8nE399Tww+K4Gr0gONLJaqNgjhOAFCT0rEyuyt/WGPzksF9Ea`
	`NBG7E0oN66zex0EOEfG1Vs6mXp1//w==`
	`-----END CERTIFICATE-----`
	`END OF INPUT`
	`Fingerprint(s): MD5`
	`Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12`
	`Do you accept this certificate? [yes/no]:`
	` yes`
		Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.
		
		The maximum number of trust points that you can authenticate to a specific CA is 10.
		
		Note For subordinate CA authentication, the Cisco NX-OS software requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export.

	Command or Action	Purpose

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto ca trustpoints** **Example:** `switch# show crypto ca trustpoints`	Displays the trust point CA information.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring Certificate Revocation Checking Methods

During security exchanges with a client (for example, an SSH user), the Cisco NX-OS device performs the certificate verification of the peer certificate sent by the client. The verification process may involve certificate revocation status checking.

You can configure the device to check the CRL downloaded from the CA. Downloading the CRL and checking locally does not generate traffic in your network. However, certificates can be revoked between downloads and your device would not be aware of the revocation.

Before you begin

Authenticate the CA.

Ensure that you have configured the CRL if you want to use CRL checking.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	
	Example:
	`switch(config)# crypto ca trustpoint`
	`admin-ca`
	`switch(config-trustpoint)#`
		Specifies a trust point CA and enters trust point configuration mode.

Step 3	**revocation-check** {**crl** [**none**] | **none**} **Example:**	Configures the certificate revocation checking methods. The default method is **crl**.

	Command or Action	Purpose
	switch(config-trustpoint)# revocation-check none	The Cisco NX-OS software uses the certificate revocation methods in the order that you specify.
Step 4	**exit** **Example:** switch(config-trustpoint)# exit switch(config)#	Exits trust point configuration mode.
Step 5	(Optional) **show crypto ca trustpoints** **Example:** switch(config)# show crypto ca trustpoints	Displays the trust point CA information.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Generating Certificate Requests

You must generate a request to obtain identity certificates from the associated trust point CA for each of your device's RSA key pairs. You must then cut and paste the displayed request into an e-mail or in a website form for the CA.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**crypto ca enroll** *name* **Example:** switch(config)# crypto ca enroll admin-ca Create the certificate request .. Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration.	Generates a certificate request for an authenticated CA. **Note** You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked.

	Command or Action	Purpose

	` Please make a note of it.`
	` Password:nbv123`
	`The subject name in the certificate will`
	`be: DeviceA.cisco.com`
	`Include the switch serial number in the`
	`subject name? [yes/no]: no`
	`Include an IP address in the subject`
	`name [yes/no]: yes`
	`ip address:172.22.31.162`
	` The certificate request will be`
	`displayed...`
	`-----BEGIN CERTIFICATE REQUEST-----`
	`MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ`
	`KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUUlDVaSMqNIgJ2kt8r14lKY`
	`0JC6ManNy4qxk8VeMKZSiLJ4JgIzKWdxbLDKTIysnjuCXGvjb+wj0hEhv/y5lT9y`
	`P2NUU8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rzargZvHtGJ91XIq4WoVkSCzXv8S`
	`VqyH0vEvAgMBAAGgITzAVBgkqhkiG9w0BCQcxCBMGnu2MITzMDYGCSqGSIb3DQEJ`
	`DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBK4WNH6IwDQYJ`
	`KoZIhvcNAQEEBQADgYEAKT60KFR6Qo8nrj0sDKZVHSfJZh6K6JtDz3Gkd99G1FWgt`
	`PftrN0WUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6U188rIIOjglXMjja8`
	`8a23bNDpNsMBrkIwA6nWkrVL8NUZEFJxdpjfngPNIZacJCUS6ZqKCMetdcKytUx0=`
	`-----END CERTIFICATE REQUEST-----`	

Step 3	**exit**
	
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 4	(Optional) **show crypto ca certificates**
	
	Example:
	`switch(config)# show crypto ca`
	`certificates`
		Displays the CA certificates.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Installing Identity Certificates

You can receive the identity certificate from the CA by e-mail or through a web browser in base64 encoded text form. You must install the identity certificate from the CA by cutting and pasting the encoded text.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	**crypto ca import** *name* **certificate**
	
	Example:
	
	```switch(config)# crypto ca import admin-ca```
	``` certificate```
	```input (cut & paste) certificate in PEM```
	```format:```
	```-----BEGIN CERTIFICATE-----```
	```MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G```
	```CSqGSIb3DQEJARYRWWlhbmRrrZLBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD```
	```VQQIEwlLYXJuYXRha2ExEEjAQBgNVBAcTCUJhbmdhbG9yZTIEOMAwGA1UEChMFQ2lz```
	```Y28xEzARBgNVBAsTCnr6ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwWXJUuYSBDQIAeFw0w```
	```NIExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEEu```
	```Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C```
	```dQlWkcjKjSICdpLifK5eJShNCQujGpzcuKsZPFXjF2UbiyeCYE8ylnrdWyw5E08rJ47```
	```glxr42/sI9IRIb/8urU/cj9jSSfKK56kroa7xWYAu8rDfz8jMChIM4WlaY/cp2q4Gb```
	```x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw```
	```GYIRVMVhYMHMS5jaXNjby5jb22HBKwiWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR```
	```bhWnlVyc9jrgMIHBBgNVHSMEgcQwgcGAFCco8kaDG6wjIEMNjsktUBoLEfnxxcYGW```
	```pIGIMIGQMSAwHgYJKoZIhvcNAQkBFhFhdWFuZGtiQGNpc2NvLmNvbTELMAkGA1UE```
	```BhMCSU4xEjAQBgNVBAgTCUthcmuGSbGFYYIESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w```
	```DAYDVQQKEwVDaXNjbzEIMBEGA1UECxMKbmV0c3RvcmFnZTIESMBAGA1UEAxMJQKBh```
	```crf5hIENBghAFYNKJrLQ21E9UEiWMrRl6MGsGA1UdHwRkMGIwLcgAsoCcgKGh0dHA6```
	```Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5janwwMKAuoCyGKtrZpbGU6```
	```Ly9cXHNZZS0wOExDXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQH```
	```AQFEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTIA4LON1crRFbnJvbGwvc3NL```
	```LIIA4XOEwXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYXZmlsZTovL1xxcc3NLLIIA4```
	```XENlcrRFbnJvbGxcc3NLLIIA4XOEwXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF```
	```AANBADoGBGSboe7GNLh9xxeOTWBNbm24U69ZSuIDoc0cUZUUIgrqnrIt qvpPyejtsyf1w```
	```E36cIZu4WsExREqxbTk8ycx7V5o=```
	```-----END CERTIFICATE-----```
		Prompts you to cut and paste the identity certificate for the CA named admin-ca.
		
		The maximum number of identify certificates that you can configure on a device is 16.

Step 3	**exit**
	
	Example:
	
	```switch(config)# exit```
	```switch#```
		Exits configuration mode.

Step 4	(Optional) **show crypto ca certificates** **Example:** ```switch# show crypto ca certificates```	Displays the CA certificates.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	
	```switch# copy running-config```
	```startup-config```
		Copies the running configuration to the startup configuration.

Ensuring Trust Point Configurations Persist Across Reboots

You can ensure that the trustpoint configuration persists across Cisco NX-OS device reboots.

The trust point configuration is a normal Cisco NX-OS device configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates, key pairs, and CRL associated with a trust point are automatically persistent if you have already copied the trust point configuration in the startup configuration. Conversely, if the trust point configuration is not copied to the startup configuration, the certificates, key pairs, and CRL associated with it are not persistent since they require the corresponding trust point configuration after a reboot. Always copy the running configuration to the startup configuration to ensure that the configured certificates, key pairs, and CRLs are persistent. Also, save the running configuration after deleting a certificate or key pair to ensure that the deletions permanent.

The certificates and CRL associated with a trust point automatically become persistent when imported (that is, without explicitly copying to the startup configuration) if the specific trust point is already saved in startup configuration.

We recommend that you create a password-protected backup of the identity certificates and save it to an external server.

Note   Copying the configuration to an external server does include the certificates and key pairs.

Exporting Identity Information in PKCS 12 Format

You can export the identity certificate along with the RSA key pair and CA certificate (or the entire chain in the case of a subordinate CA) of a trust point to a PKCS#12 file for backup purposes. You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

Note   You can use only the bootflash:*filename* format when specifying the export URL.

Before you begin

Authenticate the CA.

Install an identity certificate.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	 Example:
	 ```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	**crypto ca export** *name* **pkcs12** **bootflash:***filename password*	Exports the identity certificate and associated key pair and CA certificates for a trust point

	Command or Action	Purpose

	Example:
	
	`switch(config)# crypto ca export admin-ca`
	`pkcs12 bootflash:adminid.p12 nbv123`
		CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters.

Step 3	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	**copy booflash:***filename scheme***://***server/* [*url /*]*filename*
	
	Example:
	
	`switch# copy bootflash:adminid.p12`
	`tftp:adminid.p12`
		Copies the PKCS#12 format file to a remote server.
		
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		
		The *server*, *url*, and *filename* arguments are case sensitive.

Importing Identity Information in PKCS 12 Format

You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

> ✎
>
> **Note**   You can use only the bootflash:*filename* format when specifying the import URL.

Before you begin

Ensure that the trust point is empty by checking that no RSA key pair is associated with it and no CA is associated with the trust point using CA authentication.

Procedure

	Command or Action	Purpose

Step 1	**copy** *scheme***://** *server/*[*url /*]*filename* **bootflash:***filename*
	
	Example:
	
	`switch# copy tftp:adminid.p12`
	`bootflash:adminid.p12`
		Copies the PKCS#12 format file from the remote server.
		
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		
		The *server*, *url*, and *filename* arguments are case sensitive.

	Command or Action	Purpose

Step 2	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca import** *name* **pksc12 bootflash:***filename*
	
	Example:
	`switch(config)# crypto ca import admin-ca`
	`pkcs12 bootflash:adminid.p12 nbv123`
		Imports the identity certificate and associated key pair and CA certificates for trust point CA.

Step 4	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto ca certificates** **Example:** `switch# show crypto ca certificates`	Displays the CA certificates.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring a CRL

You can manually configure CRLs that you have downloaded from the trust points. The Cisco NX-OS software caches the CRLs in the device bootflash (cert-store). During the verification of a peer certificate, the Cisco NX-OS software checks the CRL from the issuing CA only if you have downloaded the CRL to the device and you have configured certificate revocation checking to use the CRL.

Before you begin

Ensure that you have enabled certificate revocation checking.

Procedure

	Command or Action	Purpose

Step 1	**copy** *scheme***:**[**//***server*/[*url* /]]*filename* **bootflash:***filename*
	
	Example:
	`switch# copy tftp:adminca.crl`
	`bootflash:adminca.crl`
		Downloads the CRL from a remote server.
		
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.

	Command or Action	Purpose
		The *server*, *url*, and *filename* arguments are case sensitive.

Step 2	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca crl request** *name* **bootflash:**ptar{filename}
	
	Example:
	
	`switch(config)# crypto ca crl request`
	`admin-ca bootflash:adminca.crl`
		Configures or replaces the current CRL with the one specified in the file.

Step 4	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto ca crl** *name* **Example:** `switch# show crypto ca crl admin-ca`	Displays the CA CRL information.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Deleting Certificates from the CA Configuration

You can delete the identity certificates and CA certificates that are configured in a trust point. You must first delete the identity certificate, followed by the CA certificates. After deleting the identity certificate, you can disassociate the RSA key pair from a trust point. You must delete certificates to remove expired or revoked certificates, certificates that have compromised (or suspected to be compromised) key pairs, or CAs that are no longer trusted.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name* **Example:**	Specifies a trust point CA and enters trust point configuration mode.

	Command or Action	Purpose
	`switch(config)# crypto ca trustpoint admin-ca` `switch(config-trustpoint)#`	
Step 3	**delete ca-certificate** **Example:** `switch(config-trustpoint)# delete ca-certificate`	Deletes the CA certificate or certificate chain.
Step 4	**delete certificate** [**force**] **Example:** `switch(config-trustpoint)# delete certificate`	Deletes the identity certificate. You must use the **force** option if the identity certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only the identity certificate and leave the applications (such as SSH) without a certificate to use.

Step 5	**exit**
	
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 6	(Optional) **show crypto ca certificates** [*name*] **Example:** `switch(config)# show crypto ca certificates admin-ca`	Displays the CA certificate information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting RSA Key Pairs from a Cisco NX-OS Device

You can delete the RSA key pairs from a Cisco NX-OS device if you believe the RSA key pairs were compromised in some way and should no longer be used.

Note   After you delete RSA key pairs from a device, ask the CA administrator to revoke your device's certificates at the CA. You must supply the challenge password that you created when you originally requested the certificates.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	**crypto key zeroize rsa** *label*
	
	Example:
	
	```switch(config)# crypto key zeroize rsa```
	```MyKey```
		Deletes the RSA key pair.

Step 3	**exit**
	
	Example:
	
	```switch(config)# exit```
	```switch#```
		Exits configuration mode.

Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** ```switch# show crypto key mypubkey rsa```	Displays the RSA key pair configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	
	```switch# copy running-config```
	```startup-config```
		Copies the running configuration to the startup configuration.

Verifying the PKI Configuration

To display PKI configuration information, perform one of the following tasks:

Command	**Purpose**
show crypto key mypubkey rsa	Displays information about the RSA public keys generated on the Cisco NX-OS device.
show crypto ca certificates	Displays information about CA and identity certificates.
show crypto ca crl	Displays information about CA CRLs.
show crypto ca trustpoints	Displays information about CA trust points.

Configuration Examples for PKI

This section shows examples of the tasks that you can use to configure certificates and CRLs on Cisco NX-OS devices using a Microsoft Windows Certificate server.

Note   You can use any type of certificate server to generate digital certificates. You are not limited to using the Microsoft Windows Certificate server.

Configuring Certificates on a Cisco NX-OS Device

To configure certificates on a Cisco NX-OS device, follow these steps:

Procedure

Step 1   Configure the device FQDN.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# hostname Device-1
Device-1(config)#
```

Step 2   Configure the DNS domain name for the device.

```
Device-1(config)# ip domain-name cisco.com
```

Step 3   Create a trust point.

```
Device-1(config)# crypto ca trustpoint myCA
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint: myCA; key:
revokation methods:  crl
```

Step 4   Create an RSA key pair for the device.

```
Device-1(config)# crypto key generate rsa label myKey exportable modulus 1024
Device-1(config)# show crypto key mypubkey rsa
key label: myKey
key size: 1024
exportable: yes
```

Step 5   Associate the RSA key pair to the trust point.

```
Device-1(config)# crypto ca trustpoint myCA
Device-1(config-trustpoint)# rsakeypair myKey
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint: myCA; key: myKey
revokation methods:  crl
```

Step 6   Download the CA certificate from the Microsoft Certificate Service web interface.

Step 7   Authenticate the CA that you want to enroll to the trust point.

```
Device-1(config)# crypto ca authenticate myCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRRbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
Do you accept this certificate? [yes/no]:y

Device-1(config)# show crypto ca certificates
Trustpoint: myCA
CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
serial=0560D289ACB419944F4912258CAD197A
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
purposes: sslserver sslclient ike
```

Step 8   Generate a request certificate to use to enroll with a trust point.

```
Device-1(config)# crypto ca enroll myCA
 Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password: nbv123
 The subject name in the certificate will be: Device-1.cisco.com
 Include the switch serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address: 10.10.1.1
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
```

```
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

Step 9   Request an identity certificate from the Microsoft Certificate Service web interface.

Step 10   Import the identity certificate.

```
Device-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTeu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Device-1(config)# exit
Device-1#
```

Step 11   Verify the certificate configuration.

Step 12   Save the certificate configuration to the startup configuration.

Downloading a CA Certificate

To download a CA certificate from the Microsoft Certificate Services web interface, follow these steps:

Procedure

Step 1   From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation task** and click **Next**.

Microsoft Certificate Services -- Aparna CA

Welcome

You use this web site to request a certificate for your web browser, e-mail client, or other secure program. Once you will be able to securely identify yourself to other people over the web, sign your e-mail messages, encrypt your e-mai depending upon the type of certificate you request.

Select a task:
- ⦿ Retrieve the CA certificate or certificate revocation list
- ○ Request a certificate
- ○ Check on a pending certificate

Step 2 From the display list, choose the CA certificate file to download from the displayed list. Then click **Base 64 encoded** and click **Download CA certificate**.

Microsoft Certificate Services -- Aparna CA

Retrieve The CA Certificate Or Certificate Revocation List

Install this CA certification path to allow your computer to trust certificates issued from this certification authority.

It is not necessary to manually install the CA certification path if you request and install a certificate from this certi CA certification path will be installed for you automatically.

Choose file to download:

CA Certificate: Current [Aparna CA]

○ DER encoded or ⊙ Base 64 encoded

Download CA certificate
Download CA certification path
Download latest certificate revocation list

Step 3   Click **Open** in the File Download dialog box.

Step 4   In the Certificate dialog box, click **Copy to File** and click **OK**.

Step 5   From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next**.

Step 6   In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.

Step 7   In the Certificate Export Wizard dialog box, click **Finish**.

Step 8   Enter the Microsoft Windows **type** command to display the CA certificate stored in Base-64 (PEM) format.

Requesting an Identity Certificate

To request an identify certificate from a Microsoft Certificate server using a PKCS#12 certificate signing request (CRS), follow these steps:

Procedure

Step 1   From the Microsoft Certificate Services web interface, click **Request a certificate** and click **Next**.

Step 2 Click **Advanced request** and click **Next**.

Step 3 Click **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** and click **Next**.

Microsoft Certificate Services -- Aparna CA

Advanced Certificate Requests

You can request a certificate for yourself, another user, or a computer using one of the following methods. Note that t certification authority (CA) will determine the certificates that you can obtain.

○ Submit a certificate request to this CA using a form.

⊙ Submit a certificate request using a base64 encoded PKCS #10 file or a renewal request using a base64 enco

○ Request a certificate for a smart card on behalf of another user using the Smart Card Enrollment Station.
You must have an enrollment agent certificate to submit a request for another user.

Step 4 In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next**. The certificate request is copied from the Cisco NX-OS device console.

Microsoft Certificate Services -- Aparna CA

Submit A Saved Request

Paste a base64 encoded PKCS #10 certificate request or PKCS #7 renewal request generated by an external a server) into the request field to submit the request to the certification authority (CA).

Saved Request:

Base64 Encoded
Certificate Request
(PKCS #10 or #7):

```
VqyHOvEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMG
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5j
KoZIhvcNAQEEBQADgYEAkT6OKER6Qo8njOsDXZVH
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPN
-----END CERTIFICATE REQUEST-----
```

Browse for a file to insert.

Additional Attributes:

Attributes:

Step 5   Wait one or two days until the certificate is issued by the CA administrator.

Microsoft Certificate Services -- Aparna CA

Certificate Pending

Your certificate request has been received. However, you must wait for an administrator to issue the certificate you r

Please return to this web site in a day or two to retrieve your certificate.

Note: You must return with **this** web browser within 10 days to retrieve your certificate

Step 6 Note that the CA administrator approves the certificate request.

Step 7 From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**.

Step 8 Choose the certificate request that you want to check and click **Next**.

Microsoft Certificate Services -- Aparna CA

Check On A Pending Certificate Request

Please select the certificate request you want to check:

Saved-Request Certificate (12 Nopember 2005 20:30:22)

Step 9   Click **Base 64 encoded** and click **Download CA certificate**.

Step 10   In the File Download dialog box, click **Open**.

Step 11   In the Certificate box, click **Details** tab and click **Copy to File...**. In the Certificate Export Dialog box, click
Base-64 encoded X.509 (.CER), and click **Next**.

Step 12   In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and
click **Next**.

Step 13   Click **Finish**.

Step 14 Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.

Revoking a Certificate

To revoke a certificate using the Microsoft CA administrator program, follow these steps:

Procedure

Step 1 From the Certification Authority tree, click **Issued Certificates** folder. From the list, right-click the certificate that you want to revoke.

Step 2 Choose **All Tasks > Revoke Certificate**.

Step 3 From the Reason code drop-down list, choose a reason for the revocation and click **Yes**.

Step 4 Click the **Revoked Certificates** folder to list and verify the certificate revocation.

Generating and Publishing the CRL

To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

Procedure

Step 1   From the Certification Authority screen, choose **Action > All Tasks > Publish**.

Step 2 In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.

Downloading the CRL

To download the CRL from the Microsoft CA website, follow these steps:

Procedure

Step 1   From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation list** and click **Next**.

Step 2   Click **Download latest certificate revocation list**.

Step 3 In the File Download dialog box, click **Save**.

Step 4 In the Save As dialog box, enter the destination file name and click **Save**.

Step 5 Enter the Microsoft Windows **type** command to display the CRL.

```
C:\WINNT\system32\cmd.exe                                                _ □ ×
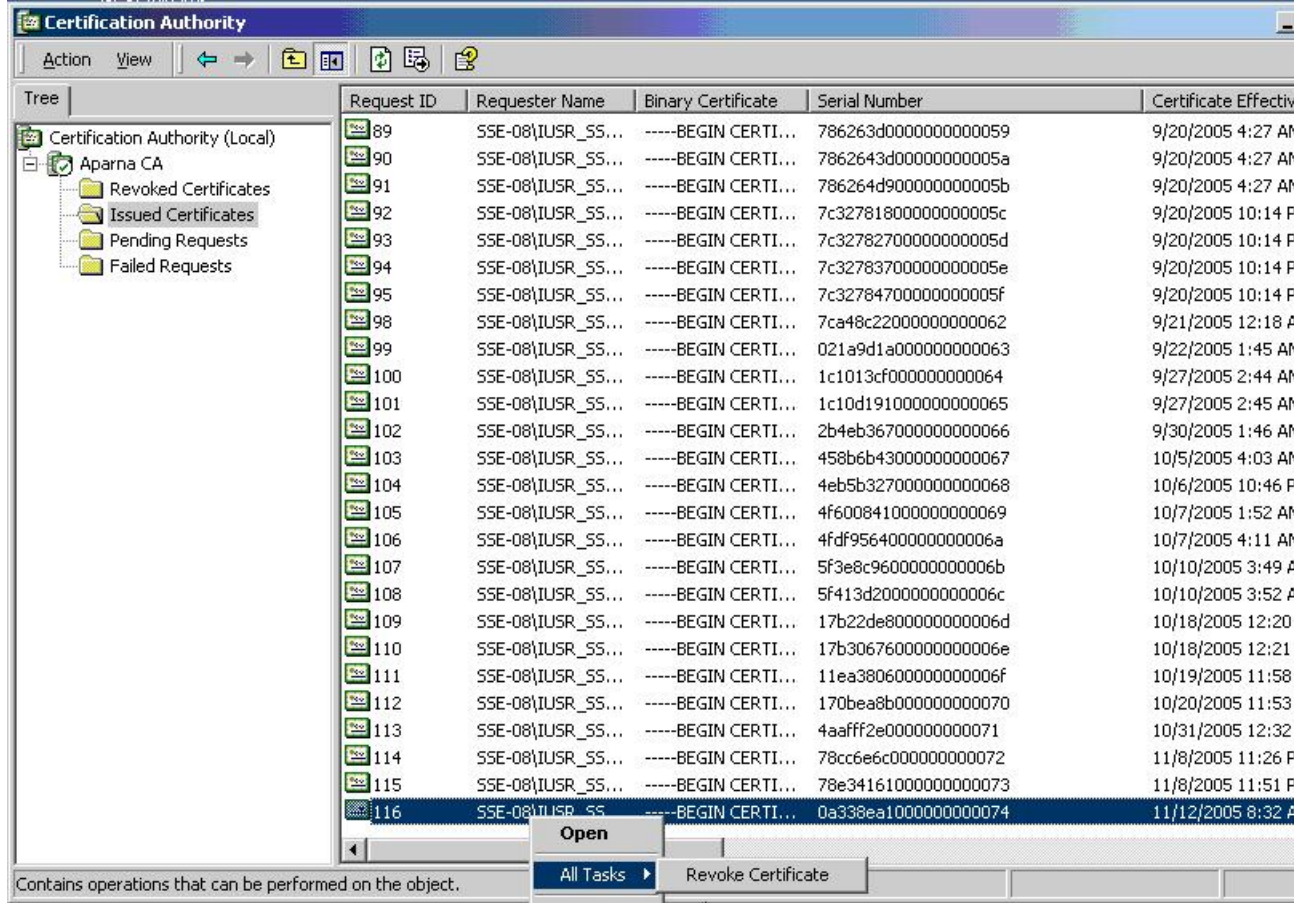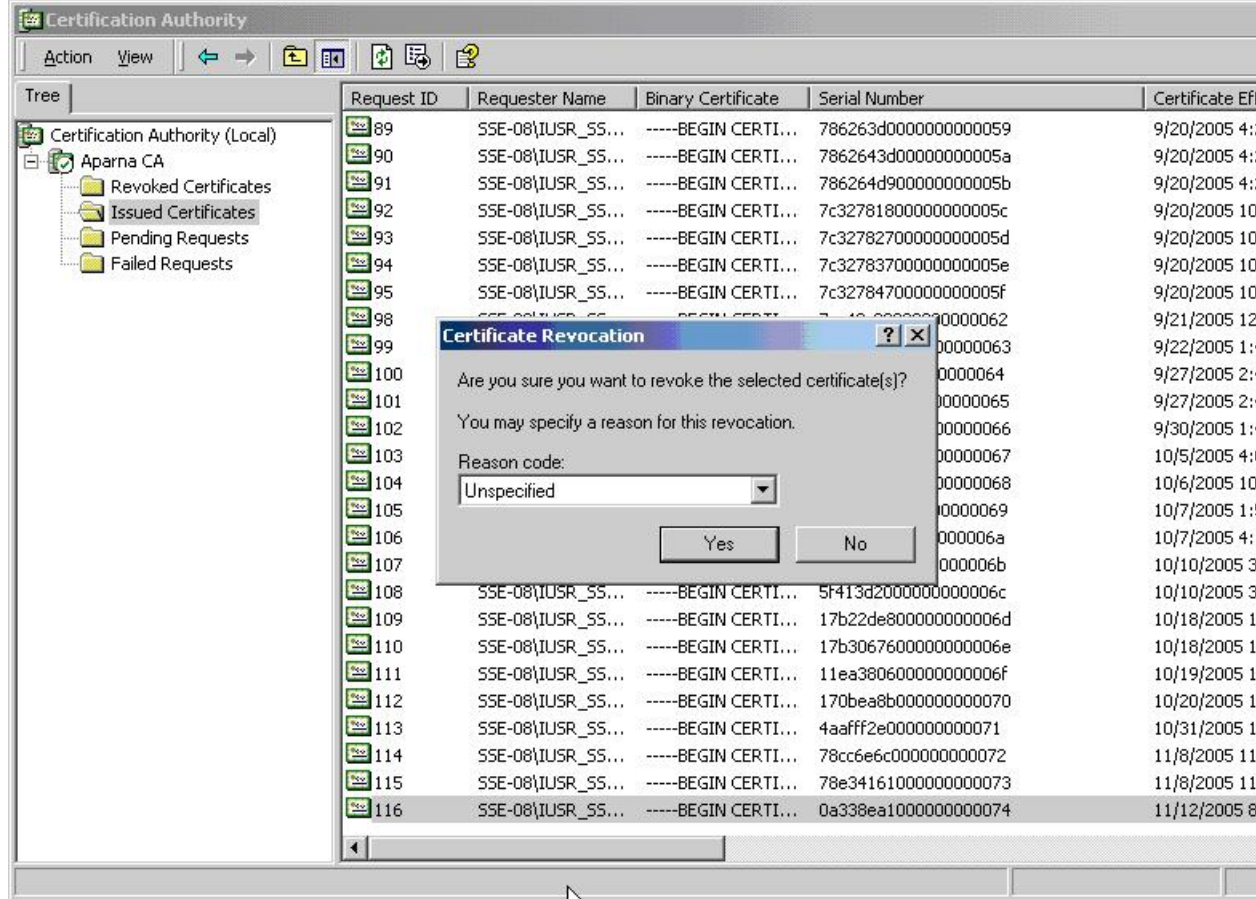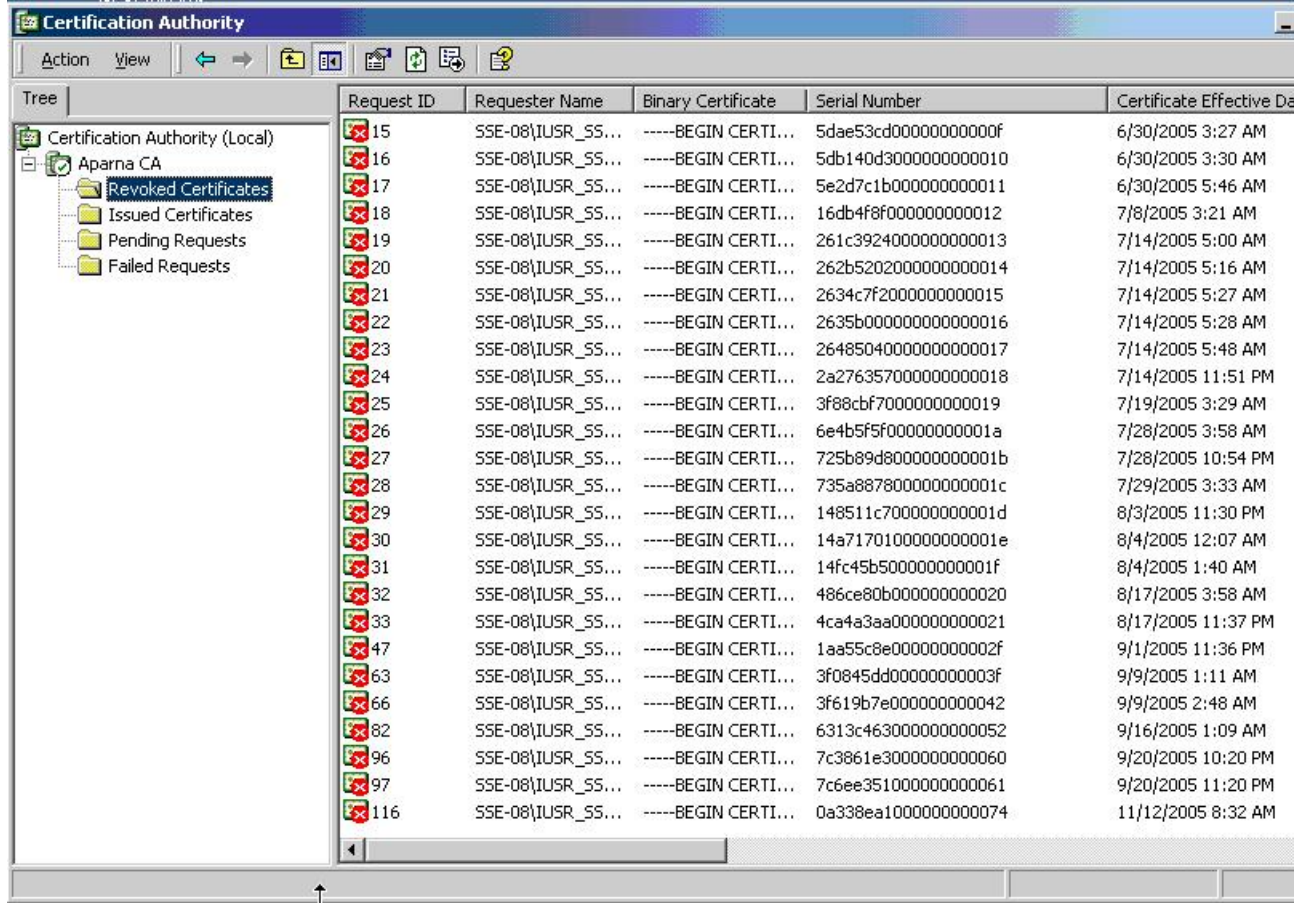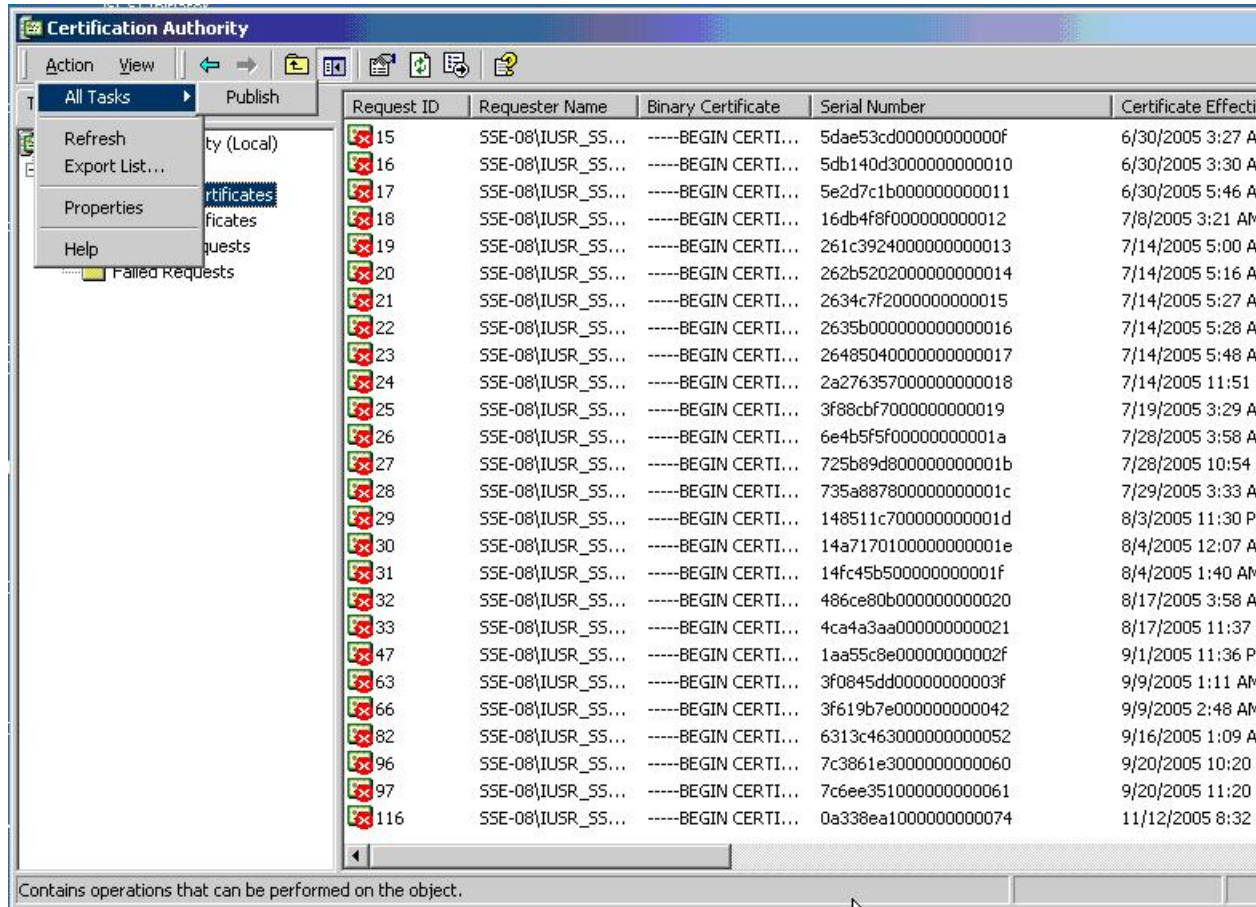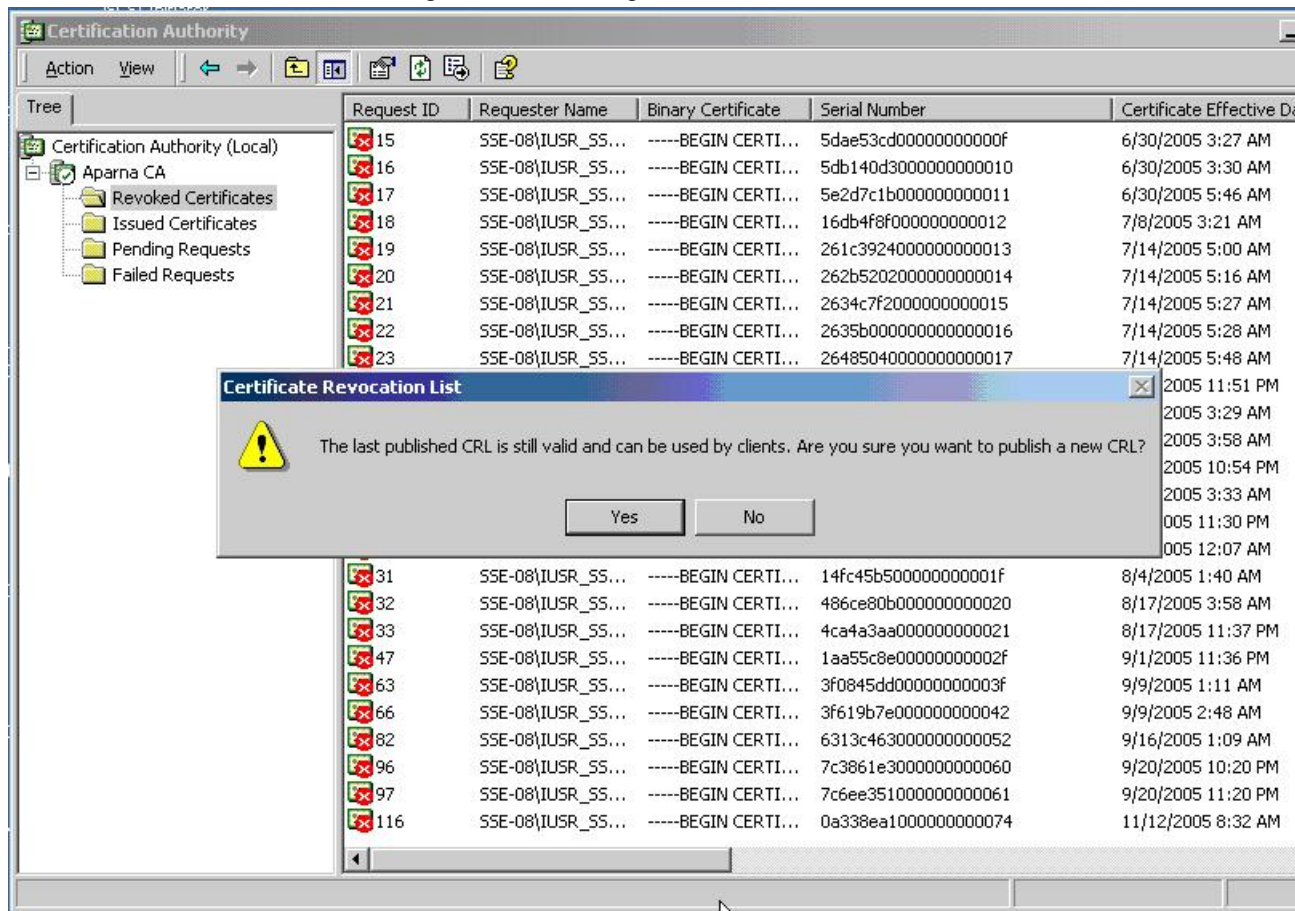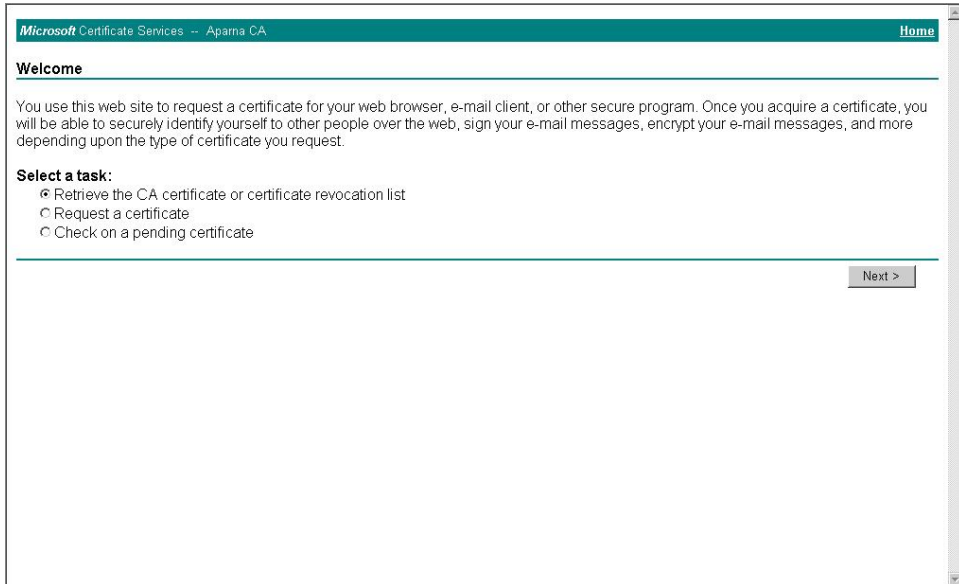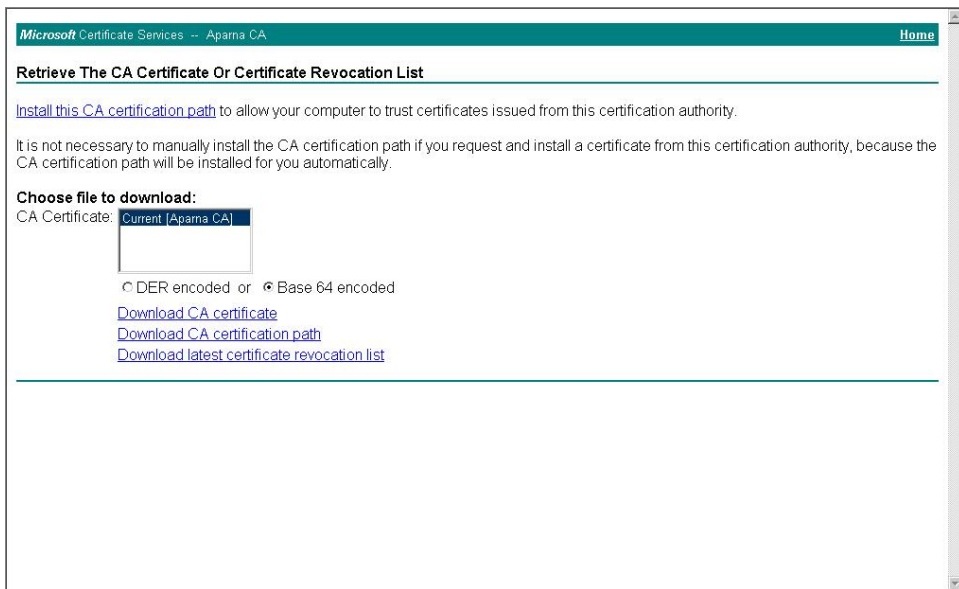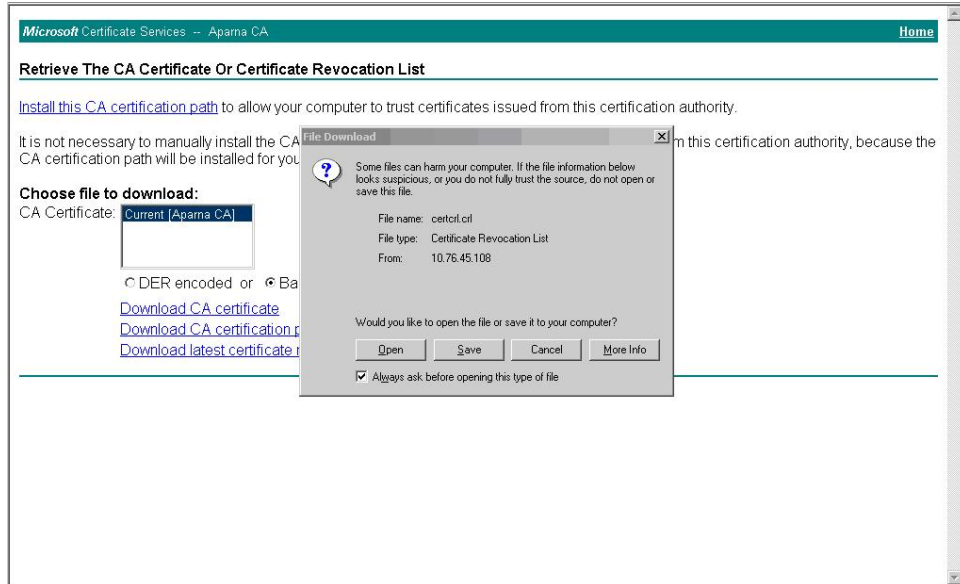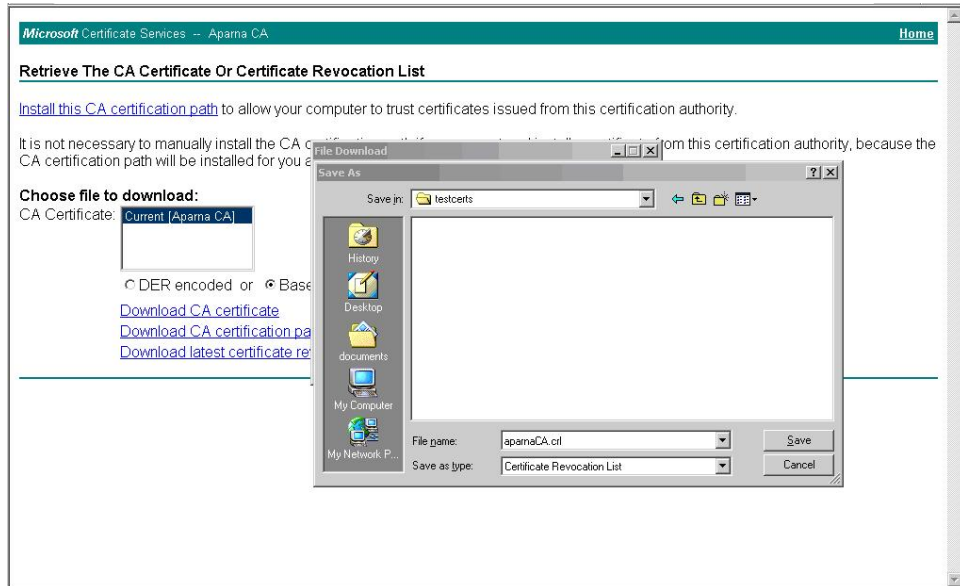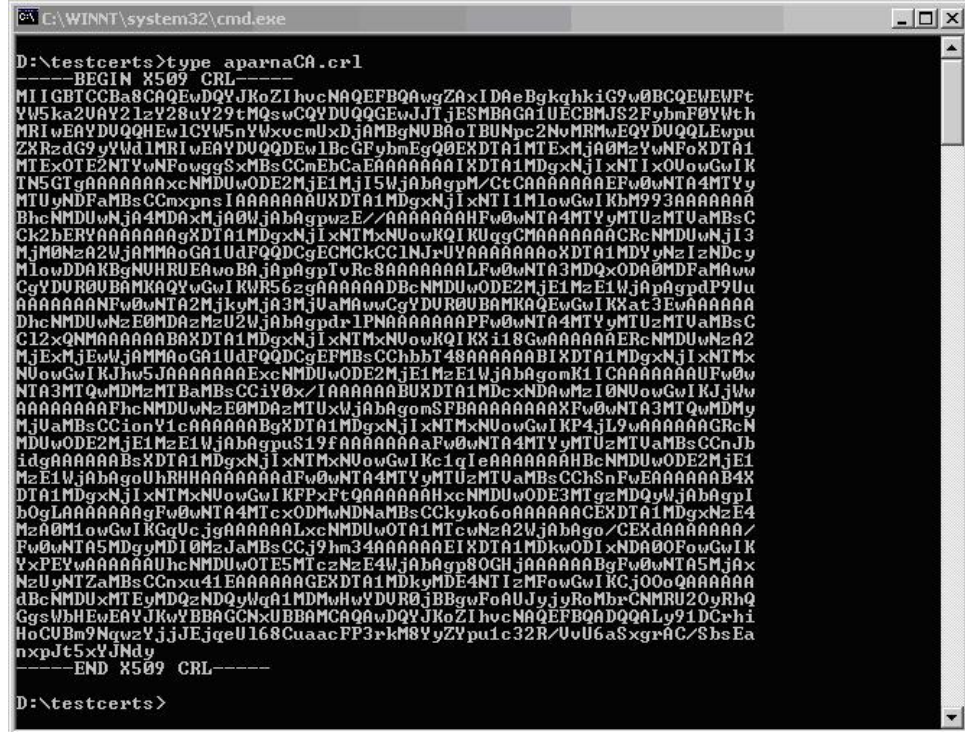
D:\testcerts>type_aparnaCA.crl
-----BEGIN X509 CRL-----
MIIGBTCCBa8CAQEwDQYJKoZIhvcNAQEFBQAwgZAxIDAeBgkqhkiG9w0BCQEWEWFt
YW5ka2UAY2lzY28uY29tMQswCQYDUQQGEwJJTjESMBAGA1UECBMJS2FybmF0YWth
MRIwEAYDUQQHEwlCYW5nYWxvcmUxDjAMBgNVBAoTBUNpc2NvMRMwEQYDUQQLEwpu
ZXRzdG9yYWdlMRIwEAYDUQQDEwlBcGFybmEgQ0ExDTIA1MTExMjA0MzYwNFoXDTA1
MTExOTE2NTYwNFowggSxMBsCCmEbCaEAAAAAAAIXDTA1MDgxNjIxNTIxOVowGwIK
TN5GTgAAAAAAxcNMDUwODE2MjE1MjI5WjAbAgpM/CtCAAAAAAAAEFw0wNTA4MTYy
MTUyNDFaMBsCCmxpnsIAAAAAAAUXDTA1MDgxNjIxNTI1MlowGwIKbM993AAAAAAA
BhcNMDUwNjA4MDAxMjA0WjAbAgpwzE//AAAAAAAHFw0wNTA4MTYyMTUzMTUaMBsC
Ck2bERYAAAAAAAgXDTA1MDgxNjIxNTMxNUowKQIKUqggCMAAAAAAACRcNMDUwNjI3
MjM0MzA2WjAMMAoGA1UdFQQDCgECMCkkCC1NJrUYAAAAAAAoXDTA1MDYyNzIzNDcy
MlowDDAKBgNVHRUEAwoBAjApAgpTvRc8AAAAAAALFw0wNTA3MDQxODA0MDFaMAww
CgYDUR0UBAMKAQYwGwIKWR56zgAAAAAADBcNMDUwODE2MjE1MzE1WjApAgpdP9Uu
AAAAAAANFw0wNTA2MjkyMjA3MjUaMAwwCgYDUR0UBAMKAQYwGwIKKXat3EwAAAAAA
DhcNMDUwNzE0MDAzMzU2U2WjApAgpdrlPNAAAAAAAAPFw0wNTA4MTYyMTUzMTVaMBsC
Cl2xQNMAAAAAAABAXDTA1MDgxNjIxNTMxNUowKQIKXii18GwAAAAAAERcNMDUwNzA2
MjExMjEwWjAMMAoGA1UdFQQDCgEFMBsCChbbI48AAAAAAABIXDTA1MDgxNjIxNTMx
NUowGwIKJhw5JAAAAAAAExcNMDUwODE2MjE1MzE1WjAbAgomK1ICAAAAAAAUFw0w
NTA3MTQwMDMzMTBaMBsCCiY0x/IAAAAAABUXDTA1MDcxNDAwMzI0NUowGwIKJjWw
AAAAAAAAFhcNMDUwNzE0MDAzMTUxWjAbAgomSFBAAAAAAAAXFw0wNTA3MTQwMDMy
MjUaMBsCCionY1cAAAAAABgXDTA1MDgxNjIxNTMxNUowGwIKP4jL9wAAAAAAAGRcN
MDUwODE2MjE1MzE1WjAbAgpuS19fAAAAAAAaFw0wNTA4MTYyMTUzMTUaMBsCCnJb
idgAAAAAABsXDTA1MDgxNjIxNTMxNUowGwIKc1qIeAAAAAAAAHBcNMDUwODE2MjE1
MzE1WjAbAgoUhRHHAAAAAAAdFw0wNTA4MTYyMTUzMTUaMBsCCChSnFwEAAAAAAB4X
DTA1MDgxNjIxNTMxNUowGwIKFPxFtQAAAAAAHxcNMDUwODE3MTgzMDQyWjAbAgpI
bOgLAAAAAAAgFw0wNTA4MTcxODMwNDNaMBsCCkyko6oAAAAAACEXDTA1MDgxNzE4
MzA0M1owGwIKGgUcjgAAAAAALxcNMDUwOTA1MTcwNzA2WjAbAgo/CEXAAAAAAA/
Fw0wNTA5MDgyMDI0MzJaMBsCCj9hm34AAAAAAAEIXDTA1MDkwODIxNDA0OFowGwIK
YxPEYwAAAAAAUhcNMDUwOTE5MTczNzE4WjAbAgp8OGHjAAAAAAABgFw0wNTA5MjAx
NzUyNTZaMBsCCnxu41EAAAAAAAGEXDTA1MDkyMDE4NTIzMFowGwIKCjOOoQAAAAAA
dBcNMDUxMTEyMDQzNDQyWjAqIMDMwHwYDUR0jBBgwFoAUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADQQALv91DCrhi
HoCUBm9NqwzYjjJEjqeU168CuaacFP3rkM8YyZYpu1c32R/VvU6aSxgrAC/SbsEa
nxpJt5xYJNdy
-----END X509 CRL-----

D:\testcerts>
```

Importing the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

Procedure

Step 1 Copy the CRL file to the Cisco NX-OS device bootflash.

```
Device-1# copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

Step 2 Configure the CRL.

```
Device-1# configure terminal
Device-1(config)# crypto ca crl request myCA bootflash:aparnaCA.crl
Device-1(config)#
```

Step 3 Display the contents of the CRL.

```
Device-1(config)# show crypto ca crl myCA
Trustpoint: myCA
CRL:
Certificate Revocation List (CRL):
        Version 2 (0x1)
```

```
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
        Last Update: Nov 12 04:36:04 2005 GMT
        Next Update: Nov 19 16:56:04 2005 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
            keyid:27:28:F2:46:83:1B:AC:23:4C:45:4D:8E:C9:18:50:1
            1.3.6.1.4.1.311.21.1:
                ...
Revoked Certificates:
    Serial Number: 611B09A1000000000002
        Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
        Revocation Date: Aug 16 21:52:29 2005 GMT
    Serial Number: 4CFC2B42000000000004
        Revocation Date: Aug 16 21:52:41 2005 GMT
    Serial Number: 6C699EC2000000000005
        Revocation Date: Aug 16 21:52:52 2005 GMT
    Serial Number: 6CCF7DDC000000000006
        Revocation Date: Jun  8 00:12:04 2005 GMT
    Serial Number: 70CC4FFF000000000007
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 4D9B1116000000000008
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 52A80230000000000009
        Revocation Date: Jun 27 23:47:06 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 5349AD460000000000A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C0000000000B
        Revocation Date: Jul  4 18:04:01 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE0000000000C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5D3FD52E0000000000D
        Revocation Date: Jun 29 22:07:25 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Key Compromise
Serial Number: 5DAB77130000000000E
        Revocation Date: Jul 14 00:33:56 2005 GMT
    Serial Number: 5DAE53CD0000000000F
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5DB140D3000000000010
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5E2D7C1B000000000011
        Revocation Date: Jul  6 21:12:10 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Cessation Of Operation
Serial Number: 16DB4F8F000000000012
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 261C3924000000000013
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 262B5202000000000014
        Revocation Date: Jul 14 00:33:10 2005 GMT
```

```
     Serial Number: 2634C7F2000000000015
          Revocation Date: Jul 14 00:32:45 2005 GMT
     Serial Number: 2635B000000000000016
          Revocation Date: Jul 14 00:31:51 2005 GMT
     Serial Number: 264850400000000000017
          Revocation Date: Jul 14 00:32:25 2005 GMT
     Serial Number: 2A276357000000000018
Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 3F88CBF7000000000019
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 6E4B5F5F00000000001A
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 725B89D800000000001B
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 735A887800000000001C
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 148511C700000000001D
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 14A7170100000000001E
          Revocation Date: Aug 16 21:53:15 2005 GMT
     Serial Number: 14FC45B500000000001F
          Revocation Date: Aug 17 18:30:42 2005 GMT
     Serial Number: 486CE80B000000000020
          Revocation Date: Aug 17 18:30:43 2005 GMT
     Serial Number: 4CA4A3AA000000000021
          Revocation Date: Aug 17 18:30:43 2005 GMT
     Serial Number: 1AA55C8E00000000002F
          Revocation Date: Sep  5 17:07:06 2005 GMT
     Serial Number: 3F0845DD00000000003F
          Revocation Date: Sep  8 20:24:32 2005 GMT
     Serial Number: 3F619B7E000000000042
          Revocation Date: Sep  8 21:40:48 2005 GMT
     Serial Number: 6313C463000000000052
          Revocation Date: Sep 19 17:37:18 2005 GMT
     Serial Number: 7C3861E3000000000060
          Revocation Date: Sep 20 17:52:56 2005 GMT
     Serial Number: 7C6EE351000000000061
          Revocation Date: Sep 20 18:52:30 2005 GMT
     Serial Number: 0A338EA1000000000074   <-- Revoked identity certificate
          Revocation Date: Nov 12 04:34:42 2005 GMT
     Signature Algorithm: sha1WithRSAEncryption
          0b:cb:dd:43:0a:b8:62:1e:80:95:06:6f:4d:ab:0c:d8:8e:32:
          44:8e:a7:94:97:af:02:b9:a6:9c:14:fd:eb:90:cf:18:c9:96:
          29:bb:57:37:d9:1f:d5:bd:4e:9a:4b:18:2b:00:2f:d2:6e:c1:
          1a:9f:1a:49:b7:9c:58:24:d7:72
```

Note   The identity certificate for the device that was revoked (serial number 0A338EA1000000000074) is listed at the end.

Configuring Access Control Lists

This chapter contains the following sections:

Information About ACLs

An access control list (ACL) is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the switch determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether the packet is permitted or denied. If there is no match, the switch applies the applicable default rule. The switch continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

IP ACL Types and Applications

The Cisco Nexus device supports IPv4 for security traffic filtering. The switch allows you to use IP access control lists (ACLs) as port ACLs, VLAN ACLs, and Router ACLs as shown in the following table.

Table 13: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	IPv4 ACLs IPv6 ACLs IPv6 ACLs with UDF-based match for Cisco NX-OS 3000 Series switches.
Router ACL	• VLAN interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	IPv4 ACLs IPv6 ACLs
VLAN ACL (VACL)	An ACL is a VACL when you use an access map to associate the ACL with an action and then apply the map to a VLAN.	IPv4 ACLs
VTY ACL	VTYs	IPv4 ACLs IPv6 ACLs

Application Order

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

1. Port ACL

2. Ingress VACL

3. Ingress Router ACL

4. Egress Router ACL

5. Egress VACL

Rules

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The switch allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host.

Protocols

IPv4, IPv6, and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 ACL, you can specify ICMP by name.

You can specify any protocol by the integer that represents the Internet protocol number.

Implicit Rules

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the switch applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the switch denies unmatched IP traffic.

All IPv6 ACLs include the following implicit rule:

```
deny ipv6 any any
```

This implicit rule ensures that the device denies unmatched IPv6 traffic.

Note
- IPv6 Neighbor Discovery packets (Neighbor Solicitation, Neighbor Advertisement, Router Solicitation, and Router Advertisement) will not be permitted due to the implicit **deny ipv6 any any** rule of an IPv6 ACL.

- You must add the following rules explicitly to allow IPv6 Neighbor Discovery packets in the Cisco Nexus N3K-C3132Q-V, N3K-C31108PC-V, N3K-C31108TC-V, N3K-C3132C-Z, N3K-C3264Q-S, N3K-C3232C, N3K-C3264C-E, N3K-C3132Q-40GE, N3K-C3172PQ-10GE, and N3K-C3172TQ-10GT platform switches:

  - **permit icmp any any nd-na**

  - **permit icmp any any nd-ns**

  - **permit icmp any any router-advertisement**

  - **permit icmp any any router-solicitation**

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. IPv4 ACLs support the following additional filtering options:

- Layer 4 protocol
- TCP and UDP ports
- ICMP types and codes
- IGMP types
- Precedence level
- Differentiated Services Code Point (DSCP) value
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections

Sequence Numbers

The Cisco Nexus device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  switch(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, the device allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers.

The Cisco Nexus device stores operator-operand couples in registers called logical operation units (LOUs) to perform operations (greater than, less than, not equal to, and range) on the TCP and UDP ports specified in an IP ACL.

Note The range operator is inclusive of boundary values.

These LOUs minimize the number of ternary content addressable memory (TCAM) entries needed to perform these operations. A maximum of two LOUs are allowed for each feature on an interface. For example an ingress RACL can use two LOUs, and a QoS feature can use two LOUs. If an ACL feature requires more than two arithmetic operations, the first two operations use LOUs, and the remaining access control entries (ACEs) get expanded.

The following guidelines determine when the device stores operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples "gt 10" and "gt 11" would be stored separately in half an LOU each. The couples "gt 10" and "lt 10" would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

You can configure the LOU threshold by using the **hardware profile tcam lou-threshold** *value* command. When the number of expanded ACEs exceeds this threshold, the device stores them in an LOU register. Otherwise, the device stores these ACEs as TCAM entries.

Note The expanded ACEs are not stored if the TCAM or all 24 LOU registers are full.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

The IPv4 TCAMs are single wide.

You can create IPv6 port ACLs, VLAN ACL, router ACLs, and you can match IPv6 addresses for QoS. However, Cisco NX-OS cannot support all of them simultaneously. You must remove or reduce the size of the existing TCAMs to enable these new IPv6 TCAMs.

TCAM region sizes have the following guidelines and limitations:

- To revert to the default ACL TCAM size, use the **no hardware profile tcam region** command. You no longer need to use the **write erase command** and reload the switch.

- Depending upon the platform, each TCAM region might have a different minimum/maximum/aggregate size restriction.

- The default size of the ARPACL TCAM is zero. Before you use the ARP ACLs in a Control Policing Plane (CoPP) policy, you must set the size of this TCAM to a non-zero size.

- You must set the VACL and egress VLAN ACL (E-VACL) size to the same value.

- Both IPv4 and IPv6 addresses cannot coexist, even in a double-wide TCAM.

- IPv6 PACL TCAM is not supported for Cisco NX-OS 3000 Series switches.

- The total TCAM depth is 2000 for ingress and 1000 for egress, which can be carved in 256 entries blocks.

- After TCAM carving, you must reload the switch.

- All existing TCAMs cannot be set to size 0.

- By default, all IPv6 TCAMs are disabled (the TCAM size is set to 0).

Table 14: TCAM Sizes by ACL Region

TCAM ACL Region	Default Size	Minimum Size	Incremental Size	Maximum Size
SUP (ingress)	128 x 2	128 x 2	N/A	128 x 2
SPAN (ingress)	128	128	N/A	128
ARPACL (ingress)	0	0	128	128
PACL (ingress)	384	ARPACL disabled = 128 ARPACL enabled = 256	256	1664 (combined)
VACL (ingress)	512	0	256	
RACL (ingress)	512	256	256	
QOS (ingress)	256	256	256	
PACL_IPV6 (ingress)	0	0	256 x 2	
VACL_IPV6 (ingress)	0	0	256 x 2	
RACL_IPV6 (ingress)	0	0	256 x 2	
QOS_IPV6 (ingress)	0	0	256 x 2	

TCAM ACL Region	Default Size	Minimum Size	Incremental Size	Maximum Size
E-VACL (egress)	512	0	256	1024 (combined)
E-RACL (egress)	512	0	256	
E-VACL_IPV6 (egress)	0	0	256 x 2	
E-RACL_IPV6 (egress)	0	0	256 x 2	
QOSLBL (pre-lookup)	256	256	256	256
SUP_IPV6 (pre-lookup)	128 x 2	256 x 2	N/A	256 x 2

Licensing Requirements for ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	No license is required to use ACLs.

Prerequisites for ACLs

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.

- You must be familiar with the interface types that you want to configure with ACLs.

VACLs have the following prerequisite:

- Ensure that the IP ACL that you want to use in the VACL exists and is configured to filter traffic in the manner that you need for this application.

Guidelines and Limitations for ACLs

IP ACLs have the following configuration guidelines and limitations:

- For all Network Forwarding Engine (NFE) enabled switches, the ingress RACLs matching the outer header of the tunnel interface are not supported.

- You cannot create ACE with SMAC or DMAC. The SMAC and DMAC options are specific to OpenFlow. OpenFlow is handled by the POLICY_MGR process and tap-aggregation is handled by the ACLMGR process. Due to this enhancement, OpenFlow specific options are not available for tap-aggregation. There is no requirement to support these options for tap-aggregation.

- You cannot configure the set-vlan option on the tap-aggregation policy. The set-vlan and strip-vlan options are specific to OpenFlow. OpenFlow and tap-aggregation are handled by two different processes. Due to this, OpenFlow specific options are not available for tap-aggregation.

- As an enhancement to HTTP method match, the tcp-option-length option has been added to the ACE syntax to specify the length of the TCP options header in the packets. You can configure up to 4 tcp-option-lengths in the ACEs, which includes the TCP option length of 0. If you do not configure the tcp-option-length option, the length is considered as 0. It means that only the packets without the TCP options header can match this ACE. This feature gives more flexibility in such a way that the HTTP method can be matched even on the packets that have the variable length TCP options header

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.

- Only 62 unique ACLs can be configured. Each ACL takes one label. If same ACL is configured on multiple interfaces, the same label is shared; but if each ACL has unique entries, the ACL labels are not shared and that label limit is 62.

- Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.

- IPv4 packets that have IP options (additional IP packet header fields following the destination address field).

- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally.

- To use the **match-local-traffic** option for all inbound and outbound traffic, you must first enable the ACL in the software.

- An RACL applied on a Layer 3 physical or logical interface does not match multicast traffic. If multicast traffic must be blocked, use a PACL instead.

- You cannot configure egress RACLs on L3 port channels.

- IPv4 ACL logging in the egress direction is not supported.

VACLs have the following configuration:

- We recommend that you perform ACL configurations using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

- ACL statistics are not supported if the DHCP snooping feature is enabled.

- If an IPv4 ACL, applied as a VLAN ACL, contains one or more ACEs with logical operators for TCP/UDP port numbers, the port numbers are matched in the ingress direction but ignored in the egress direction.

- One VLAN access map can match only one IP ACL.

- An IP ACL can have multiple permit/deny ACEs.

- One VLAN can have only one access map applied.

- The following limitations apply to Cisco Nexus 34180YC platform:

    - Egress RACL is not supported.

    - IPV6 ACLs are not supported.

    - TCAM carving is not supported . Cisco Nexus 34180YC supports two profiles – default and l3-heavy. The default has size 7166 for IP ACL and 459 for MAC ACL, while l3-heavy has size 1022 for IP ACL and size 483 for MAC ACL

    - The VACL redirect option is not supported.

    - The LOU threshold is not supported.

    - ACL Logging is not supported.

    - The **show hardware access-list tcam region** command is not supported.

Default ACL Settings

The following table lists the default settings for IP ACLs parameters.

Table 15: Default IP ACLs Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs .

The following table lists the default settings for VACL parameters.

Table 16: Default VACL Parameters

Parameters	Default
VACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs.

ACL Logging

The Cisco Nexus device supports ACL logging, which allows you to monitor flows that hit specific access control lists (ACLs). To enable the feature for the ACL entry, configure specific ACEs with the optional **log** keyword.

When you configure ACEs with the optional **log** keyword, statistics for each flow that matches the permit or deny conditions of the ACL entry are logged in the software.

ACL Consistency Checker

Beginning with Cisco NX-OS Release 9.3(3), the ACL consistency checker supports the following devices:

N3K-C3016Q-40GE, N3K-C3048TP-1GE, N3K-C3064PQ-10GE, N3K-C3064PQ-10GX, N3K-C3064T-10GT, N3K-C3132C-Z

The following entities are verified as part of the ACL consistency check:

Action, Protocol, SIP, DIP, source port, destination port, Source MAC, Destination MAC, Ethertype, COS, DSCP, VLAN and UDFs.

Cisco NX-OS supports the following PACL, RACL, and VACL consistency checker commands.

Command	Description
show consistency-checker pacl extended ingress ip module <module-id> [brief | detail]	Verifies PACL consistency check for ingress interfaces and port channel for the specified IP module.
show consistency-checker pacl extended ingress ipv6 module <module-id> [brief | detail]	Verifies PACL consistency check for ingress interfaces and port channel for the specified IPv6 module.
show consistency-checker pacl extended ingress mac module <module-id> [brief | detail]	Verifies MAC PACL consistency check for ingress interfaces and port channel for the specified MAC module.
show consistency-checker pacl extended ingress ip interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified ingress interface.
show consistency-checker pacl extended ingress ipv6 interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified IPv6 ingress interface.
show consistency-checker pacl extended ingress mac interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified ingress MAC interface.
show consistency-checker racl extended ingress ip module <module-id> [brief | detail]	Verifies RACL consistency check for ingress interfaces and port channel for the specified IP module.
show consistency-checker racl extended ingress ipv6 module <module-id> [brief | detail]	Verifies RACL consistency check for ingress interfaces and port channel for the specified IPv6 module.
show consistency-checker racl extended ingress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified ingress interface.
show consistency-checker racl extended egress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified egress IP interface.
show consistency-checker racl extended ingress ipv6 interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified ingress IPv6 interface.

Command	Description
show consistency-checker racl extended egress ipv6 interface {<int-id>|<ch-id>|<vlan-id>}[brief|detail]	Verifies RACL consistency check for the specified egress IPv6 interface.
show consistency-checker vacl extended ingress ip vlan <vlan-id> [brief	detail]	Verifies VACL consistency check for the specified IP VLAN.
show consistency-checker vacl extended ingress ipv6 vlan <vlan-id> [brief	detail]	Verifies VACL consistency check for the specified IPv6 VLAN.
show consistency-checker vacl extended ingress mac vlan <vlan-id> [brief	detail]	Verifies VACL consistency check for the specified ingress MAC VLAN.

Output Examples for ACL Consistency Checker Commands

This example shows the RACL consistency check results:

```
switch# show consistency-checker racl extended ingress ip module 1 Consistency checker
passed for Eth1/3 (ingress, ip, ip-list)
switch#
switch#
switch# show consistency-checker racl extended ingress ip module 1 brief
{
    "result": {
    "status": "CC_STATUS_OK",
    "checkers": [
      {
        "version": 1,
        "type": "CC_TYPE_IF_RACL",
        "status": "CC_STATUS_OK",
         "platformDetails": {
         "classType": "CC_PLTFM_NXOS_BCM"
       },
        "recoveryActions": [],
        "failedEntities": []
       }
     ]
    }
}
switch#
switch # show consistency-checker racl extended ingress ip interface ethernet 3/5
Consistency checker passed for Ethernet3/5 (ingress, ip, ip-list)
switch#
switch# show consistency-checker racl extended ingress ip interface ethernet 3/5 brief
{
    "result": {
    "status": "CC_STATUS_OK",
     "checkers": [
       {
        "version": 1,
        "type": "CC_TYPE_IF_RACL",
        "status": "CC_STATUS_OK",
        "platformDetails": {
        "classType": "CC_PLTFM_NXOS_BCM"
      },
        "recoveryActions": [],
        "failedEntities": []
     }
     ]
```

```
        }
    }
```

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 or IPv6 ACL on the switch and add rules to it.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for the specific Cisco Nexus device.
Step 4	(Optional) switch(config-acl)# **statistics**	Specifies that the switch maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch# **show** {**ip** | **ipv6**} **access-lists** *name*	Displays the IP ACL configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics
```

This example shows how to create an IPv6 ACL:

```
switch# configure terminal
switch(config)# ipv6 access-list acl-01-ipv6
switch(config-ipv6-acl)# permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
```

Configuring IPv4 ACL Logging

To configure the IPv4 ACL logging process, you first create the access list, then enable filtering of IPv4 traffic on an interface using the specified ACL, and finally configure the ACL logging process parameters.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ip access-list** *name*
	
	Example:
	
	`switch(config)# ip access-list`
	`logging-test`
	`switch(config-acl)#`
		Creates an IPv4 ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3	{**permit**	**deny**} **ip** *source-address destination-address* **log**
		
		Example:
		
		`switch(config-acl)# permit ip any`
		`10.30.30.0/24 log`
			Creates an ACL rule that permits or denies IPv4 traffic matching its conditions. To enable the system to generate an informational logging message about each packet that matches the rule, you must include the **log** keyword.
			
			The *source-address* and *destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, or **any** to designate any address.

Step 4	**exit**
	
	Example:
	
	`switch(config-acl)# exit`
	`switch(config)#`
		Updates the configuration and exits IP ACL configuration mode.

Step 5	**interface ethernet** *slot/port*
	
	Example:
	
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 6	**ip access-group** *name* **in**
	
	Example:
	
	`switch(config-if)# ip access-group`
	`logging-test in`
		Enables the filtering of IPv4 traffic on an interface using the specified ACL. You can apply an ACL to inbound traffic.

	Command or Action	Purpose

Step 7	**exit**
	
	Example:
	
	`switch(config-if)# exit`
	`switch(config)#`
		Updates the configuration and exits interface configuration mode.

Step 8	**logging ip access-list cache interval** *interval* **Example:** `switch(config)# logging ip access-list cache interval 490`	Configures the log-update interval (in seconds) for the ACL logging process. The default value is 300 seconds. The range is from 5 to 86400 seconds.
Step 9	**logging ip access-list cache entries** *number-of-flows* **Example:** `switch(config)# logging ip access-list cache entries 8001`	Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576.
Step 10	**logging ip access-list cache threshold** *threshold* **Example:** `switch(config)# logging ip access-list cache threshold 490`	If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message.
Step 11	**hardware rate-limiter access-list-log** *packets* **Example:** `switch(config)# hardware rate-limiter access-list-log 200`	Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000.
Step 12	**acllog match-log-level** *severity-level* **Example:** `switch(config)# acllog match-log-level 5`	Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging).
Step 13	(Optional) **show logging ip access-list cache** [**detail**] **Example:** `switch(config)# show logging ip access-list cache`	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces. No other information of active flows will be displayed specifically all the unsupported options.

Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip**	**ipv6**} **access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 4	switch(config-acl)# [*sequence-number*] {**permit**	**deny**} *protocol source destination*	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 5	(Optional) switch(config-acl)# **no** {*sequence-number*	{**permit**	**deny**} *protocol source destination*}	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 6	(Optional) switch(config-acl)# [**no**] **statistics**	Specifies that the switch maintains global statistics for packets that match the rules in the ACL. The **no** option stops the switch from maintaining global statistics for the ACL.
Step 7	(Optional) switch#**show ip access-lists** *name*	Displays the IP ACL configuration.
Step 8	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 180

Removing an IP ACL

You can remove an IP ACL from the switch.

Before you remove an IP ACL from the switch, be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not

affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no** {**ip**	**ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration. The removed IP ACL should not appear.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	(Optional) switch# **show** {**ip**	**ipv6**} **access-lists** *name*	Displays the IP ACL configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL to mgmt0

You can apply an IPv4 or IPv6 ACL to the management interface (mgmt0).

Before you begin

Ensure that the ACL that you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface mgmt** *port*
	
	Example:
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Enters configuration mode for the management interface.

Step 3	**ip access-group** *access-list* {**in** | **out**}
	
	Example:
	`switch(config-if)#ip access-group acl-120`
	`out`
		Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.

Step 4	(Optional) **show running-config aclmgr**
	
	Example:
	`switch(config-if)# show running-config`
	`aclmgr`
		Displays the ACL configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

• Creating an IP ACL

Applying an IP ACL as a Port ACL

You can apply an IPv4 ACL to a physical Ethernet interface or a PortChannel. ACLs applied to these interface types are considered port ACLs.

✎

Note   Some configuration parameters when applied to an PortChannel are not reflected on the configuration of the member ports.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** [*chassis/*]*slot/port* | **port-channel** *channel-number*}	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **ip port access-group** *access-list* **in**	Applies an IPv4 ACL to the interface or PortChannel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Logical operation units (LOUs) are not available for router ACLs applied in the out direction. If an IPv4 ACL is applied as a router ACL in the out direction, access control entries (ACEs) that contain logical operators for TCP/UDP port numbers are expanded internally to multiple ACEs and might require more TCAM entries when compared to the same ACL applied in the in direction.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port*[**.** *number*] • switch(config)# **interface port-channel** *channel-number*[**.** *number*] • switch(config)# **interface tunnel** *tunnel-number* • switch(config)# **interface vlan** *vlan-ID* • switch(config)# **interface mgmt** *port*	Enters configuration mode for the interface type that you specified.
Step 3	Enter one of the following commands: • switch(config-if)# **ip access-group** *access-list* {**in** | **out**} • switch(config-if)# **ipv6 traffic-filter** *access-list* {**in** | **out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Verifying the IP ACL Configuration

To display IP ACL configuration information, perform one of the following tasks.

Command	Purpose
show hardware access-list tcam region	Displays the TCAM sizes that will be applicable on the next reload of the device.

Command	Purpose
show hardware access-list tcam template {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}	Displays the configuration for all TCAM templates or for a specific template. **nfe**—The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series, 3164Q, and 31128PQ devices. **nfe2**—The default TCAM template for NFE2-enabled Cisco Nexus 9500, 3232C, and 3264Q devices. **l2-l3**—The default TCAM template for Layer 2 and Layer 3 configurations on Cisco Nexus 9200 Series switches. **l3**—The default TCAM template for Layer 3 configurations on Cisco Nexus 9200 Series switches.
show ip access-lists	Displays the IPv4 ACL configuration.
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show logging ip access-list cache [**detail**]	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, and source interfaces. No other information of active flows will be displayed specifically all the unsupported options.
show logging ip access-list status	Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value.
show running-config acllog	Displays the ACL log running configuration.

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied.
	Note This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config acllog	Displays the ACL log startup configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration.
	Note This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Monitoring and Clearing IP ACL Statistics

Use the **show ip access-lists** or **show ipv6 access-list** command to display statistics about an IP ACL, including the number of packets that have matched each rule. For detailed information about the fields in the output from this command, see the *Command Reference* for your Cisco Nexus device.

Note   The mac access-list is applicable to non-IPv4 and non-IPv6 traffic only.

Procedure

- switch# **show** {**ip** | **ipv6**} **access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** and **show ipv6 access-list** command output includes the number of packets that have matched each rule.

- switch#**show ip access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.

- switch# **clear** {**ip** | **ipv6**} **access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

- switch# **clear ip access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

Triggering the RACL Consistency Checker

You can manually trigger the RACL consistency checker to compare the hardware and software configuration of the ingress and egress RACLs of a module and display the results. To manually trigger the RACL consistency checker and display the results, use the following command in any mode:

Procedure

	Command or Action	Purpose
Step 1	**show consistency-checker racl module** *slot*	Starts an RACL consistency check on the specified module and displays the results.

Example

This example shows how to trigger an RACL consistency check and display the results:

```
switch# show consistency-checker racl module 1
Validates RACL on up interfaces:
Consistency Check: FAILED

Found consistencies for following Interfaces:
   Ethernet1/9 (in)
   Ethernet1/9 (out)
   Ethernet1/17 (in)
   Ethernet1/17 (out)

Found inconsistencies for following Interfaces and EID:
   Ethernet1/3 (in)
   Ethernet1/3 (out)
```

Configuring ACL Logging

Configuring the ACL Logging Cache

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **logging ip access-list cache entries** *num_entries*	Sets the maximum number of log entries cached in the software. The range is from 0 to 1000000 entries. The default value is 8000 entries.
Step 3	switch(config)# **logging ip access-list cache interval** *seconds*	Sets the number of seconds between log updates. If an entry is inactive for this duration, it is removed from the cache. The range is from 5 to 86400 seconds. The default value is 300 seconds.
Step 4	switch(config)# **logging ip access-list cache threshold** *num_packets*	Sets the number of packet matches before an entry is logged. The range is from 0 to 1000000 packets. The default value is 0 packets, which means that logging is not triggered by the number of packet matches.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example show how to set the maximum number of log entries to 5000, the interval to 120 seconds, and the threshold to 500000:

```
switch# configure terminal
switch(config)# logging ip access-list cache entries 5000
switch(config)# logging ip access-list cache interval 120
switch(config)# logging ip access-list cache threshold 500000
switch(config)# copy running-config startup-config
```

Applying ACL Logging to an Interface

You can apply ACL logging to Ethernet interfaces and port channels.

Before you begin

- Create an ACL.

- Create an IP access list with at least one access control entry (ACE) configured for logging.

- Configure the ACL logging cache.

- Configure the ACL log match level.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies the Ethernet interface.
Step 3	switch(config-if)# **ip access-group** *name* **in**	Attaches an ACL with a log to the specified interface. ACL logging is enabled when the ACL is applied to the interface on the hardware.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to apply the Ethernet interface with the logging specified in acl1 for all ingress traffic:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip access-group acl1 in
switch(config-if)# copy running-config startup-config
```

Applying the ACL Log Match Level

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **acllog match-log-level** *number*	Specifies the logging level to match for entries to be logged in the ACL log (acllog). The number is a value from 0 to 7. The default is 6.
		Note Log messages are entered into the log if the logging level for the ACL log facility (acllog) and the logging severity level for the log file are greater than or equal to the ACL log match log level setting.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to apply the log match level for entries to be logged in the ACL log:

```
switch# configure terminal
switch(config)# acllog match-log-level 3
switch(config)# copy running-config startup-config
```

Clearing Log Files

You can clear messages in the log file and the NVRAM.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear logging ip access-list cache**	Clears the access control list (ACL) cache.

Verifying the ACL Logging Configuration

To display ACL logging configuration information, perform one of the following tasks:

Command	Purpose
switch# **show startup-config acllog**	Displays the access control list (ACL) log file in the startup configuration.
switch# **show running-config acllog**	Displays the access control list (ACL) log file in the running configuration.
switch# **show logging ip access-list cache**	Displays the IP access list cache.
switch# **show logging ip access-list cache detail**	Displays detailed information about the IP access list cache.
switch# **show logging ip access-list status**	Displays the status of the IP access list cache.

Configuring ACL Using HTTP Methods to Redirect Requests

Starting with Release 6.0(2)U5(1), the HTTP method option is added to the ACL CLI. You can intercept and redirect specific HTTP methods to a server that is connected to a specific port.

Note   As an enhancement to HTTP method match, the tcp-option-length option has been added to the ACE syntax to specify the length of the TCP options header in the packets. You can configure up to 4 tcp-option-lengths in the ACEs, which includes the TCP option length of 0. If you do not configure the tcp-option-length option, the length is considered as 0. It means that only the packets without the TCP options header can match this ACE. This feature gives more flexibility in such a way that the HTTP method can be matched even on the packets that have the variable length TCP options header.

The following HTTP methods can be redirected:

- connect

- delete

- get

- head

- post

- put

- trace

Configure the ACL CLI to redirect specific HTTP methods to a server.

Before you begin

- Create an IP access list.

- Enable the double wide TCAM for the IFACL region using the CLI **hardware profile tcam region ifacl 512 double-wide** command . This command applies to the global configuration and only on Trident2 based Cisco Nexus 3000 Series switches. Reload the switch for this configuration to take into effect.

- Enable tap-aggregation feature to redirect the packets to another interface using the CLI **hardware profile tap-aggregation** command. This command applies to global configuration. Reload the switch for this configuration to take into effect.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# **permit** *protocol source* any http-method ? **Example:** ```switch(config-acl)# permit tcp 1.1.1.1/32 any http-method ? connect Match http packets with```	Configures the ACL CLI to redirect specific HTTP methods to a server.

	Command or Action	Purpose
| | ```
CONNECT method [0x434f4e4e]
  delete   Match http packets with DELETE
 method [0x44454c45]
  get      Match http packets with GET
method [0x47455420]
  head     Match http packets with HEAD
 method [0x48454144]
  post     Match http packets with POST
 method [0x504f5354]
  put      Match http packets with PUT
method [0x50555420]
  trace    Match http packets with TRACE
 method [0x54524143]
``` | |
| **Step 4** | (Optional) switch# **show ip access-lists** *name* | Displays the IP ACL configuration. |
| **Step 5** | (Optional) switch# **show run interface** *<x/y>* | Displays the interface configuration. |

Example

In the following example, an Ethernet interface 1/33 is receiving HTTP traffic. Ethernet interface 1/34 is the egress interface. Enable mode **tap-aggregation** on the egress interface. Create an ACL to match the traffic. Configure the redirect HTTP get method that matches the ACL to Ethernet interface 1/34. Apply the ACL to the port where the HTTP traffic is received. Any HTTP get traffic that hits the ACL on Ethernet 1/33 is redirected to the destination interface, for example, Ethernet 1/34. The same steps can be used for the other listed methods.

Troubleshooting Information—In case the ACL is not hit or the packets are not redirected, ensure that double wide TCAM is enabled. Ensure that tap aggregation is enabled. Ensure both source and destination ports are in STP forwarding state in the same VLAN. Ensure that the ACL is programmed in TCAM using the **sh platform afm info attachment interface <interface>** command. The HTTP redirect feature does not work on Layer 3 ports.

Note Use the CLI command **show hardware access-list interface <>** for checking the ACL policy under the interface.

```
switch# configure terminal
switch(config)# interface Ethernet 1/33

L3-QI2-CR-one(config)## interface Ethernet 1/34
L3-QI2-CR-one(config-if)# mode tap-aggregation
switch(config)# ip access-list  http-redirect-acl
switch(config-acl)# 10 permit tcp 10.1.1.1/32 10.2.2.2/32 http-method get redirect e1/34
switch(config-acl)# 10 permit tcp any any http-method get tcp-option-length 8 redirect e1/34
switch(config-acl)# 20 permit tcp any any http-method post redirect e1/34
switch(config-acl)# statistics per-entry

switch(config)# interface Ethernet 1/33
switch(config-if)# ip port access-group http-redirect-acl in

switch(config)# show ip access-lists
switch(config)# show run int 1/34
switch(config)# show hardware access-list interface 1/34
```

Information About VLAN ACLs

A VLAN ACL (VACL) is one application of an IP ACL. You can configure VACLs to apply to all packets that are bridged within a VLAN. VACLs are used strictly for security packet filtering. VACLs are not defined by direction (ingress or egress).

VACLs and Access Maps

VACLs use access maps to link an IP ACL to an action. The switch takes the configured action on packets that are permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.

- Drop—Drops the traffic.

Statistics

The Cisco Nexus device can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

> **Note** The Cisco Nexus device does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the switch maintains statistics for that VACL. This allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

Configuring VACLs

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP ACL with an action to be applied to the matching traffic.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **vlan access-map** *map-name*	Enters access map configuration mode for the access map specified.
Step 3	switch(config-access-map)# **match ip address** *ip-access-list*	Specifies an IPv4 and IPv6 ACL for the map.
Step 4	switch(config-access-map)# **action** {**drop** | **forward**}	Specifies the action that the switch applies to traffic that matches the ACL.
Step 5	(Optional) switch(config-access-map)# [**no**] **statistics**	Specifies that the switch maintains global statistics for packets matching the rules in the VACL. The **no** option stops the switch from maintaining global statistics for the VACL.
Step 6	(Optional) switch(config-access-map)# **show running-config**	Displays the ACL configuration.
Step 7	(Optional) switch(config-access-map)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

Be sure that you know whether the VACL is applied to a VLAN. The switch allows you to remove VACLs that are current applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the switch considers the removed VACL to be empty.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no vlan access-map** *map-name*	Removes the VLAN access map configuration for the specified access map.
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **vlan filter** *map-name* **vlan-list** *list*	Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. The **vlan-list** command can specify a list of up to 32 VLANs, but multiple **vlan-list** commands can be configured to cover more than 32 VLANs.
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Verifying VACL Configuration

To display VACL configuration information, perform one of the following tasks:

Command or Action	Purpose
switch# **show running-config aclmgr**	Displays ACL configuration, including VACL-related configuration.
switch# **show vlan filter**	Displays information about VACLs that are applied to a VLAN.
switch#**show vlan access-map**	Displays information about VLAN access maps.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, perform one of the following tasks:

Procedure

- switch# **show vlan access-list**

  Displays VACL configuration. If the VLAN access-map includes the **statistics** command, then the **show vlan access-list** command output includes the number of packets that have matched each rule.

- switch# **clear vlan access-list counters**

  Clears statistics for all VACLs or for a specific VACL.

Configuration Examples for VACL

The following example shows how to configure a VACL to forward traffic permitted by an IP ACL named acl-ip-01 and how to apply the VACL to VLANs 50 through 82:

```
switch# configure terminal

switch(config)# vlan access-map acl-ip-map

switch(config-access-map)# match ip address acl-ip-01

switch(config-access-map)# action forward

switch(config-access-map)# exit

 switch(config)# vlan filter acl-ip-map vlan-list 50-82
```

Configuring the LOU Threshold

You can configure the LOU threshold. When the number of expanded ACEs exceeds this threshold, the device stores them in an LOU register. Otherwise, the device stores these ACEs as TCAM entries. This configuration takes effect only for the next ACL configuration. All existing ACL configurations either in TCAM or LOU register are not affected by this configuration. In order for the changes to take effect, you have to use the **copy r s** command and reload the box.

> **Note** The expanded ACEs are not stored if the TCAM or all 24 LOU registers are full.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam lou-threshold** *value*	Configures the LOU threshold and the LOU expansion threshold takes effect for the new policies. It is recommended to save the configuration and reload so that the threshold takes effect on the already existing policies.
		The threshold values range from 1 to 100, and the default LOU threshold value is 1.

Example

This example shows how to configure the LOU threshold:

```
switch# configure terminal
switch(config)# hardware profile tcam lou-threshold 20
switch(config)# copy running-config startup-config
switch(config)# reload
LOU expansion threshold changed to 20
```

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hardware profile tcam region** {**arpacl** | {**ipv6-e-racl** | **e-racl**} | **ifacl** | | {**ipv6-qos** | **qos**} |**qoslbl** | {**ipv6-racl** | **racl**} | {**ipv6-span** | **span** } | {**ipv6-span-l2** | **span** } | {**spanv6** |**span** } | {**spanv6-12** |**span**} | **vacl** } {**fhs**} *tcam_size*	Changes the ACL TCAM region size. • **arpacl**—Configures the size of the Address Resolution Protocol (ARP) ACL (ARPACL) TCAM region. • **e-racl**—Configures the size of the egress router ACL (ERACL) TCAM region. • **e-vacl**—Configures the size of the egress VLAN ACL (EVACL) TCAM region. • **ifacl**—Configures the size of the interface ACL (ifacl) TCAM region. The maximum number of entries is 1500. • **qos**—Configures the size of the quality of service (QoS) TCAM region. • **qoslbl**—Configures the size of the QoS Label (qoslbl) TCAM region. • **racl**—Configures the size of the router ACL (RACL) TCAM region. • **span**—Configures the size of the SPAN TCAM region. • **span-l2**—Configures the size of the Layer 2 SPAN TCAM region. • **spanv6**—Configures the SPAN TCAM region. • **spanv6-12**—Configures the Layer 2 SPAN TCAM region. • **vacl**—Configures the size of the VLAN ACL (VACL) TCAM region. • **fhs**—Configures the size of the fhs TCAM region.

	Command or Action	Purpose
		• *tcam_size*—TCAM size. The range is from 0 to 2,14,74, 83, 647 entries. For FHS, the range is from 0-4096.
		Note **vacl** and **e-vacl** TCAM regions should be set to the same size.

Step 3	**copy running-config startup-config**
	
	 Example:
	
	 `switch(config)# copy running-config`
	 `startup-config`
		Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Step 4	switch(config)# **show hardware profile tcam region**
	
	 Example:
	
	 `switch(config)# show hardware profile`
	 `tcam region`
		Displays the TCAM sizes that will be applicable on the next reload of the switch.

Step 5	switch(config)# **reload** **Example:** `switch(config)# reload`	Copies the running configuration to the startup configuration. **Note** The new size values are effective only upon the next reload after saving the **copy running-config to startup-config**.

Example

The following example shows how to change the size of the RACL TCAM region:

```
switch(config)# hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

The following example shows the error message you see when you set the ARP ACL TCAM value to a value other than 0 or 128, and then shows how to change the size of the ARP ACL TCAM region:

```
switch(config)# hardware profile tcam region arpacl 200
ARPACL size can be either 0 or 128

switch(config)# hardware profile tcam region arpacl 128
To start using ARPACL tcam, IFACL tcam size needs to be changed.
Changing IFACL tcam size to 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
```

The following example shows how to configure the TCAM VLAN ACLs on a switch:

```
switch# configure sync
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-sync)# switch-profile s5010
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)# hardware profile tcam region vacl 512
switch(config-sync-sp)# hardware profile tcam region e-vacl 512
switch(config-sync-sp)#
```

The following example shows how to qualify the UDF on IPv6 SPAN:

```
switch(config)# hardware profile  tcam  region  ipv6-span  512
Warning: Please save config and reload the system for the configuration to take effect
switch(config)# hardware  profile  tcam  region  spanv6 qualify  udf  udf1
[SUCCESS] Changes to UDF qualifier set will be applicable only after reboot.
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware profile tcam region
         sup size =   16
        vacl size =  640
       ifacl size =  496
         qos size =  256
       rbacl size =    0
        span size =    0
        racl size = 1536
      e-racl size =  256
      e-vacl size =  640
      qoslbl size =    0
       arpacl size =    0
   ipv6-racl size =    0
 ipv6-e-racl size =    0
     ipv6-sup size =    0
     ipv6-qos size =    0
```

Note   On Cisco Nexus 3000 Series switches, you must carve the switch RACL TCAM regions in order to make IGMP and PIM work on Layer 3 interfaces. Some system default Multicast ACLs that are installed in the RACL regions are required for IGMP and PIM to work on Layer 3 interfaces.

Note   On Cisco Nexus 3000 Series switches, you musst carve the switch IPv6 SPAN/IPv6 SPAN-L2 TCAM regions inorder to make the IPv6 UDFs work on the Layer 2 and the Layer 3 ports respectively.

Note   If the config-control property is set to YES in the XML hierarchy definition file, then it is possible for the memory object to use a faulty bit map to report the error.

Reverting to the Default TCAM Region Sizes

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** ``` switch# configure terminal switch(config)# ```	Enters global configuration mode.
Step 2	switch(config)# **no hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Reverts the configuration to the default ACL TCAM size.
Step 3	(Optional) **copy running-config startup-config** **Example:** ``` switch(config)# copy running-config startup-config ```	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **reload**	Reloads the switch.

Example

The following example shows how to revert to the default RACL TCAM region sizes:

```
switch(config)# no hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-configur startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configuring ACLs on Virtual Terminal Lines

To restrict incoming and outgoing connections for IPv4 or IPv6 between a Virtual Terminal (VTY) line and the addresses in an access list, use the **access-class** command in line configuration mode. To remove access restrictions, use the **no** form of this command.

Follow these guidelines when configuring ACLs on VTY lines:

- Set identical restrictions on all VTY lines because a user can connect to any of them.

- Statistics per entry is not supported for ACLs on VTY lines.

Before you begin

Be sure that the ACL that you want to apply exists and is configured to filter traffic for this application.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **line vty**
	
	Example:
	
	`switch(config)# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

Step 3	switch(config-line)# **access-class access-list-number {in | out}**
	
	Example:
	
	`switch(config-line)# access-class ozi2`
	`in`
	`switch(config-line)#access-class ozi3`
	`out`
	`switch(config)#`
		Specifies inbound or outbound access restrictions.

Step 4	(Optional) switch(config-line)# **no access-class access-list-number {in | out}**
	
	Example:
	
	`switch(config-line)# no access-class ozi2`
	`in`
	`switch(config-line)# no access-class ozi3`
	`out`
	`switch(config)#`
		Removes inbound or outbound access restrictions.

Step 5	switch(config-line)# **exit**
	
	Example:
	
	`switch(config-line)# exit`
	`switch#`
		Exits line configuration mode.

Step 6	(Optional) switch# **show running-config aclmgr** **Example:** `switch# show running-config aclmgr`	Displays the running configuration of the ACLs on the switch.

Step 7	(Optional) switch# **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Example

The following example shows how to apply the access-class ozi2 command to the in-direction of the vty line.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# line vty
switch(config-line)# access-class ozi2 in
switch(config-line)# exit
switch#
```

Verifying ACLs on VTY Lines

To display the ACL configurations on VTY lines, perform one of the following tasks:

Command	Purpose
show running-config aclmgr	Displays the running configuration of the ACLs configured on the switch.
show users	Displays the users that are connected.
show access-lists *access-list-name*	Display the statistics per entry.

Configuration Examples for ACLs on VTY Lines

The following example shows the connected users on the console line (ttyS0) and the VTY lines (pts/0 and pts/1).

```
switch# show users
NAME     LINE         TIME          IDLE          PID COMMENT
admin    ttyS0        Aug 27 20:45  .             14425 *
admin    pts/0        Aug 27 20:06 00:46          14176 (172.18.217.82) session=ssh
admin    pts/1        Aug 27 20:52  .             14584 (10.55.144.118)
```

The following example shows how to allow vty connections to all IPv4 hosts except 172.18.217.82 and how to deny vty connections to any IPv4 host except 10.55.144.118, 172.18.217.79, 172.18.217.82, 172.18.217.92:

```
switch# show running-config aclmgr
!Time: Fri Aug 27 22:01:09 2010
version 5.0(2)N1(1)
ip access-list ozi
  10 deny ip 172.18.217.82/32 any
  20 permit ip any any
ip access-list ozi2
  10 permit ip 10.55.144.118/32 any
  20 permit ip 172.18.217.79/32 any
  30 permit ip 172.18.217.82/32 any
  40 permit ip 172.18.217.92/32 any


line vty
  access-class ozi in
  access-class ozi2 out
```

The following example shows how to configure the IP access list by enabling per-entry statistics for the ACL:

```
switch# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
switch(config)# ip access-list ozi2
switch(config-acl)# statistics per-entry
switch(config-acl)# deny tcp 172.18.217.83/32 any
```

```
switch(config-acl)# exit

switch(config)# ip access-list ozi
switch(config-acl)# statistics per-entry
switch(config-acl)# permit ip 172.18.217.20/24 any
switch(config-acl)# exit
switch#
```

The following example shows how to apply the ACLs on VTY in and out directions:

```
switch(config)# line vty
switch(config-line)# ip access-class ozi in
switch(config-line)# access-class ozi2 out
switch(config-line)# exit
switch#
```

The following example shows how to remove the access restrictions on the VTY line:

```
switch# configure terminal
Enter configuration commands, one per line. End
with CNTL/Z.
switch(config)# line vty
switch(config-line)# no access-class ozi2 in
switch(config-line)# no ip access-class ozi2 in
switch(config-line)# exit
switch#
```

Configuring Port Security

This chapter contains the following sections:

About Port Security

Port security allows you to configure Layer 2 physical interfaces and Layer 2 port-channel interfaces to allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

Note Unless otherwise specified, the term *interface* refers to both physical interfaces and port-channel interfaces; likewise, the term *Layer 2 interface* refers to both Layer 2 physical interfaces and Layer 2 port-channel interfaces.

Secure MAC Address Learning

The process of securing a MAC address is called learning. A MAC address can be a secure MAC address on one interface only. For each interface on which you enable port security, the device can learn a limited number

of MAC addresses by the static or dynamic methods. The way that the device stores secure MAC addresses varies depending upon how the device learned the secure MAC address.

Static Method

The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are unaffected if the device restarts.

A static secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address from the configuration.

- You configure the interface to act as a Layer 3 interface.

Adding secure addresses by the static method is not affected by whether dynamic address learning is enabled.

Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- The device restarts

- The interface restarts

- The address reaches the age limit that you configured for the interface

- You explicitly remove the address

- You configure the interface to act as a Layer 3 interface

Sticky Method

If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning, but the device stores addresses learned by this method in nonvolatile RAM (NVRAM). As a result, addresses learned by the sticky method persist through a device restart. Sticky secure MAC addresses do not appear in the running configuration of an interface.

Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, the device stops dynamic learning and performs sticky learning instead. If you disable sticky learning, the device resumes dynamic learning.

A sticky secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address

- You configure the interface to act as a Layer 3 interface

Dynamic Address Aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 0 to 1440 minutes, where 0 disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The only method of determining address age is:

Absolute

The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Note When the absolute aging time is configured, MAC aging occurs even when the traffic from the source MAC is flowing. However, during MAC aging and re-learn, there could be a transient traffic drop.

Secure MAC Address Maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: static or dynamic.

Tip To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

The following three limits can determine how many secure MAC addresses are permitted on an interface:

Device Maximum

The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

Interface Maximum

You can configure a maximum number of 1025 secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Interface maximums cannot exceed the device maximum.

VLAN Maximum

You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. A VLAN maximum cannot exceed the configured interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first.

Security Violations and Actions

Port security triggers security violations when either of the following events occurs:

MAC Count Violation

Ingress traffic arrives at an interface from a nonsecure MAC address, and learning the address would exceed the applicable maximum number of secure MAC addresses.

When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

- VLAN 1 has a maximum of five addresses

- The interface has a maximum of ten addresses

The device detects a violation when any of the following occurs:

- The device has learned five addresses for VLAN 1, and inbound traffic from a sixth address arrives at the interface in VLAN 1.

- The device has learned ten addresses on the interface, and inbound traffic from an eleventh address arrives at the interface.

The possible actions that the device can take are as follows:

Shutdown

Shuts down the interface that received the packet triggering the violation. The interface is error disabled. This action is the default. After you reenable the interface, it retains its port security configuration, including its secure MAC addresses.

You can use the **errdisable** global configuration command to configure the device to reenable the interface automatically if a shutdown occurs, or you can manually reenable the interface by entering the **shutdown** and **no shutdown** interface configuration commands.

Restrict

Drops ingress traffic from any nonsecure MAC addresses.

The device keeps a count of the number of dropped MAC addresses, which is called the security violation count. Address learning continues until the maximum security violations have occurred on the interface. Traffic from addresses learned after the first security violation is dropped.

MAC Move Violation

Ingress traffic from a secure MAC address arrives at a different interface in the same VLAN as the interface on which the address is secured.

You see a mac move notification only when the the logging level of Layer2 Forwarding Module (L2FM) is increased to 4 or 5

When a MAC move violation occurs, the device increments the security violation counter for the interface, and irrespective of the violation mode configured, the interface is error disabled. If the violation mode is configured as Restrict or Protect, the violation is logged in the system log.

Because a MAC move violation results in the interface being error disabled, irrespective of the violation mode configured, we recommend using the **errdisable** command to enable automatic errdisable recovery.

Port Security and Port Types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

Access Ports

You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN. VLAN maximums are not useful for access ports.

Trunk Ports

You can configure port security on interfaces that you have configured as Layer 2 trunk ports. The device allows VLAN maximums only for VLANs associated with the trunk port.

SPAN Ports

You can configure port security on SPAN source ports but not on SPAN destination ports.

Ethernet Port Channels

You can configure port security on Layer 2 Ethernet port channels in either access mode or trunk mode.

Note Port security is supported for FEX interfaces only in non-vPC deployments on Cisco Nexus 9300-EX/FX/FX2/FX3 Series switches. Beginning with Cisco NX-OS Release 9.3(5), Nexus 9300-FX3 Series switches are supported.

Port Security and Port-Channel Interfaces

Port security is supported on Layer 2 port-channel interfaces. Port security operates on port-channel interfaces in the same manner as on physical interfaces, except as described in this section.

General Guidelines

Port security on a port-channel interface operates in either access mode or trunk mode. In trunk mode, the MAC address restrictions enforced by port security apply to all member ports on a per-VLAN basis.

Enabling port security on a port-channel interface does not affect port-channel load balancing.

Port security does not apply to port-channel control traffic passing through the port-channel interface. Port security allows port-channel control packets to pass without causing security violations. Port-channel control traffic includes the following protocols:

- Port Aggregation Protocol (PAgP)

- Link Aggregation Control Protocol (LACP)

- Inter-Switch Link (ISL)

- IEEE 802.1Q

Configuring Secure Member Ports

The port security configuration of a port-channel interface has no effect on the port security configuration of member ports.

Adding a Member Port

If you add a secure interface as a member port of a port-channel interface, the device discards all dynamic secure addresses learned on the member port but retains all other port-security configuration of the member port in the running configuration. Static secure MAC addresses learned on the secure member port are also stored in the running configuration rather than NVRAM.

If port security is enabled on the member port and not enabled on the port-channel interface, the device warns you when you attempt to add the member port to the port-channel interface. You can use the **force** keyword with the **channel-group** command to forcibly add a secure member port to a nonsecure port-channel interface.

While a port is a member of a port-channel interface, you cannot configure port security on the member port. To do so, you must first remove the member port from the port-channel interface.

Removing a Member Port

If you remove a member port from a port-channel interface, the device restores the port security configuration of the member port. Static secure MAC addresses that were learned on the port before you added it to the port-channel interface are restored to NVRAM and removed from the running configuration.

Note To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

Removing a Port-Channel Interface

If you remove a secure port-channel interface, the following occurs:

- The device discards all secure MAC addresses learned for the port-channel interface, including static secure MAC addresses learned on the port-channel interface.

- The device restores the port-security configuration of each member port. The static secure MAC addresses that were learned on member ports before you added them to the port-channel interface are restored to NVRAM and removed from the running configuration. If a member port did not have port security enabled prior to joining the port-channel interface, port security is not enabled on the member port after the port-channel interface is removed.

Note To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

Disabling Port Security

If port security is enabled on any member port, the device does not allow you to disable port security on the port-channel interface. To do so, remove all secure member ports from the port-channel interface first. After disabling port security on a member port, you can add it to the port-channel interface again, as needed.

Port Type Changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

Access Port to Trunk Port

When you change a Layer 2 interface from an access port to a trunk port, the device drops all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN.

Switched Port to Routed Port

When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

Routed Port to Switched Port

When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

Licensing Requirements for Port Security

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Port security requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for Port Security

Port security has the following prerequisites:

- You must globally enable port security for the device that you want to protect with port security.

Default Settings for Port Security

This table lists the default settings for port security parameters.

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown

Guidelines and Limitations for Port Security

When configuring port security, follow these guidelines:

- Port security does not support switched port analyzer (SPAN) destination ports.

- Port security does not depend upon other features.

- Port security is not supported on PVLAN and VXLAN interfaces.

- Port security is not supported on vPCs. Also, we do not recommend enabling port security in vPC deployments.

- When static MAC addresses are added using the command **mac address-table static** *mac-address* **vlan** *vlan-ID* **interface** *interface-name* and port security is enabled on the given interface, the configured static MAC addresses will be deleted from the system.

- Port Security is not supported on the Cisco Nexus 34180YC platform.

Guidelines and Limitations for Port Security on vPC

Apart from the guidelines and limitations for port security, check that you can meet the following guidelines and limitations for port security on vPCs:

- You must enable port security globally on both vPC peers in a vPC domain.

- You must enable port security on the vPC interfaces of both vPC peers.

- You must configure a static secure MAC address on the primary vPC peer. The static MAC address is synchronized with the secondary vPC peer. You can also configure a static secure MAC address on the secondary peer. The second static MAC address appears in the secondary vPC configuration but does not take affect.

- You must ensure that the maximum MAC count value remains the same for both primary and secondary vPC ports.

- On a secondary vPC port, there is no limit check for static MACs configured. Cisco recommends that you configure the same number of static MACs on a secondary vPC port as defined in the maximum MAC count.

- All learned MAC addresses are synchronized between vPC peers.

- Both vPC peers can be configured using the dynamic or static MAC address learning method. Cisco recommends that you configure both vPC peers using the same method. This helps prevent port shut down (errDisabled state) in certain cases, such as a vPC role change.

- Dynamic MAC addresses are dropped only after the age limit is reached on both vPC peers.

- You set the maximum number of secure MAC addresses on the primary vPC switch. The primary vPC switch does the count validation and disregards any maximum number settings on the secondary switch.

- You must configure the violation action on the primary vPC. When a security violation is triggered, the security action defined on the primary vPC switch occurs.

- You can use the **show vpc consistency-parameters id** command to verify that the configuration is correct on both vPC peers.

- While a switch undergoes an in-service software upgrade (ISSU), port security operations are stopped on its peer switch. The peer switch does not learn any new MAC addresses, and MAC moves occurring during this operation are ignored. When the ISSU is complete, the peer switch is notified and normal port security functionality resumes.

- ISSU to higher versions is supported; however, ISSU to lower versions is not supported.

Configuring Port Security

Enabling or Disabling Port Security Globally

You can enable or disable port security globally on a device. By default, port security is disabled globally.

When you disable port security, all port security configuration on the interface is ineffective. When you disable port security globally, all port security configuration is lost.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature port-security** **Example:** `switch(config)# feature port-security`	Enables port security globally. The **no** option disables port security globally.
Step 3	(Optional) **show port-security** **Example:** `switch(config)# show port-security`	Displays the status of port security.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

When you disable port security on an interface, all switchport port security configuration for the interface is lost.

Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the Ethernet or port-channel interface that you want to configure with port security.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

Step 4	[**no**] **switchport port-security**
	
	Example:
	
	`switch(config-if)# switchport`
	`port-security`
		Enables port security on the interface. The **no** option disables port security on the interface.

Step 5	(Optional) **show running-config port-security**
	
	Example:
	
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Enabling or Disabling Sticky MAC Address Learning

You can disable or enable sticky MAC address learning on an interface. If you disable sticky learning, the device returns to dynamic MAC address learning on the interface, which is the default learning method.

By default, sticky MAC address learning is disabled.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with sticky MAC address learning.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

Step 4	[**no**] **switchport port-security mac-address sticky**
	
	Example:
	`switch(config-if)# switchport`
	`port-security mac-address sticky`
		Enables sticky MAC address learning on the interface. The **no** option disables sticky MAC address learning.

Step 5	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Adding a Static Secure MAC Address on an Interface

You can add a static secure MAC address on a Layer 2 interface.

✎

Note   If the MAC address is a secure MAC address on any interface, you cannot add it as a static secure MAC address to another interface until you remove it from the interface on which it is already a secure MAC address.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

You must have enabled port security globally.

Verify that the interface maximum has not been reached for secure MAC addresses. If needed, you can remove a secure MAC address, or you can change the maximum number of addresses on the interface.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you specify.

Step 3	[**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
	
	Example:
	
	`switch(config-if)# switchport`
	`port-security mac-address 0019.D2D0.00AE`
		Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Removing a Static Secure MAC Address on an Interface

You can remove a static secure MAC address on a Layer 2 interface.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a static secure MAC address.

Step 3	**no switchport port-security mac-address** *address*
	
	Example:
	`switch(config-if)# no switchport`
	`port-security mac-address 0019.D2D0.00AE`
		Removes the static secure MAC address from port security on the current interface.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Removing a Sticky Secure MAC Address

You can remove a sticky secure MAC address, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	
	```switch(config)# interface ethernet 2/1```
	```switch(config-if)#```
		Enters interface configuration mode for the interface from which you want to remove a sticky secure MAC address.

Step 3	**no switchport port-security mac-address sticky**
	
	Example:
	
	```switch(config-if)# no switchport```
	```port-security mac-address sticky```
		Disables sticky MAC address learning on the interface, which converts any sticky secure MAC addresses on the interface to dynamic secure MAC addresses.

Step 4	**clear port-security dynamic address** *address*
	
	Example:
	
	```switch(config-if)# clear port-security```
	```dynamic address 0019.D2D0.02GD```
		Removes the dynamic secure MAC address that you specify.

Step 5	(Optional) **show port-security address interface** {**ethernet** *slot*/*port* | **port-channel** *channel-number*}
	
	Example:
	
	```switch(config)# show port-security```
	```address interface ethernet 2/1```
		Displays secure MAC addresses. The address that you removed should not appear.

Step 6	(Optional) **switchport port-security mac-address sticky**
	
	Example:
	
	```switch(config-if)# switchport```
	```port-security mac-address sticky```
		Enables sticky MAC address learning again on the interface.

Removing a Dynamic Secure MAC Address

You can remove dynamically learned, secure MAC addresses.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**clear port-security dynamic** {**interface ethernet** *slot*/*port* | **address** *address*} [**vlan** *vlan-ID*]
	
	Example:
	`switch(config)# clear port-security`
	`dynamic interface ethernet 2/1`
		Removes dynamically learned, secure MAC addresses, as specified.
		
		If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify.
		
		If you use the **address** keyword, you remove the single, dynamically learned address that you specify.
		
		Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.

Step 3	(Optional) **show port-security address**
	
	Example:
	`switch(config)# show port-security`
	`address`
		Displays secure MAC addresses.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of addresses is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Note   When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the device rejects the command. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	   • **interface ethernet** *slot*/*port*
	   • **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot* is the interface that you want to configure with the maximum number of MAC addresses.

Step 3	[**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]
	
	Example:
	`switch(config-if)# switchport`
	`port-security maximum 425`
		Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 1025. The **no** option resets the maximum number of MAC addresses to the default, which is 1.
		
		If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring an Address Aging Type and Time

You can configure the MAC address aging type and the length of time that the device uses to determine when MAC addresses learned by the dynamic method have reached their age limit.

Absolute aging is the default aging type.

By default, the aging time is 0 minutes, which disables aging.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	  • **interface ethernet** *slot*/*port*
	  • **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with the MAC aging type and time.

Step 3	[**no**] **switchport port-security aging type absolute**
	
	Example:
	`switch(config-if)# switchport`
	`port-security aging type absolute`
		Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging.

Step 4	[**no**] **switchport port-security aging time** *minutes*
	
	Example:
	`switch(config-if)# switchport`
	`port-security aging time 120`
		Configures the number of minutes that a dynamically learned MAC address must age before the device drops the address. The maximum valid *minutes* is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).

Step 5	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring a Security Violation Action

You can configure the action that the device takes if a security violation occurs. The violation action is configurable on each interface that you enable with port security.

The default security action is to shut down the port on which the security violation occurs.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	
	    • **interface ethernet** *slot*/*port*
	    • **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with a security violation action.

Step 3	[**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**}
	
	Example:
	`switch(config-if)# switchport`
	`port-security violation restrict`
		Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Verifying the Port Security Configuration

To display the port security configuration information, perform one of the following tasks.

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status of the device.
show port-security interface	Displays the port security status of a specific interface.

Command	Purpose
show port-security address	Displays secure MAC addresses.
show vpc consistency-parameters vpc id	Verifies configuration on both vPC peers.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses.

Configuration Example for Port Security

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

Configuration Examples for Port Security in a vPC Domain

The following example shows how to enable and configure port security on vPC peers in a vPC domain. The first switch is the primary vPC peer and the second switch is the secondary vPC peer. Before configuring port security on the switches, create the vPC domain and check that the vPC peer-link adjacency is established.

Additional References for Port Security

Related Documents

Related Topic	Document Title
Layer 2 switching	*Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide*

MIBs

Cisco NX-OS provides read-only SNMP support for port security.

MIBs	MIBs Link
CISCO-PORT-SECURITY-MIB	To locate and download MIBs, go to the following URL:
Note Traps are supported for notification of secure MAC address violations.	https://snmp.cloudapps.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2

Configuring DHCP Snooping

This chapter contains the following sections:

Information About DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.

- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.

- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

Feature Enabled and Globally Enabled

When you are configuring DHCP snooping, it is important that you understand the difference between enabling the DHCP snooping feature and globally enabling DHCP snooping.

Feature Enablement

The DHCP snooping feature is disabled by default. When the DHCP snooping feature is disabled, you cannot configure it or any of the features that depend on DHCP snooping. The commands to configure DHCP snooping and its dependent features are unavailable when DHCP snooping is disabled.

When you enable the DHCP snooping feature, the switch begins building and maintaining the DHCP snooping binding database. Features dependent on the DHCP snooping binding database can now make use of it and can therefore also be configured.

Enabling the DHCP snooping feature does not globally enable it. You must separately enable DHCP snooping globally.

Disabling the DHCP snooping feature removes all DHCP snooping configuration from the switch. If you want to disable DHCP snooping and preserve the configuration, globally disable DHCP snooping but do not disable the DHCP snooping feature.

Global Enablement

After DHCP snooping is enabled, DHCP snooping is globally disabled by default. Global enablement is a second level of enablement that allows you to have separate control of whether the switch is actively performing DHCP snooping that is independent from enabling the DHCP snooping binding database.

When you globally enable DHCP snooping, on each untrusted interface of VLANs that have DHCP snooping enabled, the switch begins validating DHCP messages that are received and used the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

When you globally disable DHCP snooping, the switch stops validating DHCP messages and validating subsequent requests from untrusted hosts. It also removes the DHCP snooping binding database. Globally disabling DHCP snooping does not remove any DHCP snooping configuration or the configuration of other features that are dependent upon the DHCP snooping feature.

Trusted and Untrusted Sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source might initiate traffic attacks or other hostile actions. To prevent such attacks, DHCP snooping filters messages from untrusted sources.

In an enterprise network, a trusted source is a switch that is under your administrative control. These switches include the switches, routers, and servers in the network. Any switch beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any switch that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In a Cisco Nexus device, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to switches (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note   For DHCP snooping to function properly, you must connect all DHCP servers to the switch through trusted interfaces.

DHCP Snooping Binding Database

Using information extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts that are connected through trusted interfaces.

Note   The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

DHCP snooping updates the database when the switch receives specific DHCP messages. For example, the feature adds an entry to the database when the switch receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the switch receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

Information About the DHCPv6 Relay Agent

DHCPv6 Relay Agent

You can configure the device to run a DHCPv6 relay agent, which forwards DHCPv6 packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCPv6 messages and then generate a new DHCPv6 message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCPv6 packet) and forwards it to the DHCPv6 server.

VRF Support for the DHCPv6 Relay Agent

You can configure the DHCPv6 relay agent to forward DHCPv6 broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCPv6 servers in a different VRF. By using a single DHCPv6 server to provide DHCPv6 support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF.

Licensing Requirements for DHCP Snooping

This feature does not require a license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for DHCP Snooping

You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent .

Guidelines and Limitations for DHCP Snooping

Consider the following guidelines and limitations when configuring DHCP snooping:

- The same MAC address is permitted in the static DHCP binding across multiple IP and ports.

- The DHCP snooping database can store 2000 bindings.

- DHCP snooping is not active until you enable the feature, enable DHCP snooping globally, and enable DHCP snooping on at least one VLAN.

- Before globally enabling DHCP snooping on the switch, make sure that the switches that act as the DHCP server and the DHCP relay agent are configured and enabled.

- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

- DHCP snooping and DHCP relay feature are not supported on the same VLAN.

- DHCP snooping should not be followed by DHCP relay in the network (DHCP snooping does not work when the DHCP relay is configured on the same Nexus device).

- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.

- For DHCPv6 relay, you can configure up to 32 DHCPv6 server addresses on an interface.

Default Settings for DHCP Snooping

This table lists the default settings for DHCP snooping parameters.

Table 17: Default DHCP Snooping Parameters

Parameters	Default
DHCP snooping feature	Disabled
DHCP snooping globally enabled	No

Parameters	Default
DHCP snooping VLAN	None
DHCP snooping Option 82 support	Disabled
DHCP snooping trust	Untrusted
VRF support for the DHCP relay agent	Disabled
VRF support for the DHCPv6 relay agent	Disabled
DHCP relay agent	Disabled
DHCPv6 relay agent	Disabled
DHCPv6 relay option type cisco	Disabled

Configuring DHCP Snooping

Minimum DHCP Snooping Configuration

1. Enable the DHCP snooping feature.

2.

Procedure

	Command or Action	Purpose
Step 1	Enable the DHCP snooping feature.	When the DHCP snooping feature is disabled, you cannot configure DHCP snooping. For details, see Enabling or Disabling the DHCP Snooping Feature, on page 227.
Step 2	Enable DHCP snooping globally.	For details, see Enabling or Disabling DHCP Snooping Globally, on page 228.
Step 3	Enable DHCP snooping on at least one VLAN.	By default, DHCP snooping is disabled on all VLANs. For details, see Enabling or Disabling DHCP Snooping on a VLAN, on page 229.
Step 4	Ensure that the DHCP server is connected to the switch using a trusted interface.	For details, see Configuring an Interface as Trusted or Untrusted, on page 231.

Enabling or Disabling the DHCP Snooping Feature

You can enable or disable the DHCP snooping feature on the switch. By default, DHCP snooping is disabled.

Before you begin

If you disable the DHCP snooping feature, all DHCP snooping configuration is lost. If you want to turn off DHCP snooping and preserve the DHCP snooping configuration, disable DHCP globally.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature dhcp** **Example:** `switch(config)# feature dhcp`	Enables the DHCP snooping feature. The **no** option disables the DHCP snooping feature and erases all DHCP snooping configuration.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping Globally

You can enable or disable the DHCP snooping globally on the switch. Globally disabling DHCP snooping stops the switch from performing any DHCP snooping but preserves DCHP snooping configuration.

Before you begin

Ensure that you have enabled the DHCP snooping feature. By default, DHCP snooping is globally disabled.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** option disables DHCP snooping.

	Command or Action	Purpose
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs.

Before you begin

By default, DHCP snooping is disabled on all VLANs.

Ensure that DHCP snooping is enabled.

> **Note** If a VACL is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list*
	
	Example:
	
	`switch(config)# ip dhcp snooping vlan`
	`100,200,250-252`
		Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP snooping on the VLANs specified.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

Command or Action	Purpose
switch(config)# copy running-config startup-config	

Enabling or Disabling Option 82 Data Insertion and Removal

You can enable or disable the insertion and removal of Option 82 information for DHCP packets forwarded without the use of the DHCP relay agent. By default, the device does not include Option 82 information in DHCP packets.

Note DHCP relay agent support for Option 82 is configured separately.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ip dhcp snooping information option** **Example:** switch(config)# ip dhcp snooping information option	Enables the insertion and removal of Option 82 information for DHCP packets. The **no** option disables the insertion and removal of Option 82 information.
Step 3	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Enabling or Disabling Strict DHCP Packet Validation

You can enable or disable the strict validation of DHCP packets by the DHCP snooping feature. By default, strict validation of DHCP packets is disabled.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**[no] ip dhcp packet strict-validation**
	
	Example:
	`switch(config)# ip dhcp packet`
	`strict-validation`
		Enables the strict validation of DHCP packets by the DHCP snooping feature. The **no** option disables strict DHCP packet validation.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring an Interface as Trusted or Untrusted

You can configure whether an interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces

- Layer 2 port-channel interfaces

Before you begin

By default, all interfaces are untrusted.

Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: • **interface ethernet** *port*/*slot*	• Enters interface configuration mode, where *port* / *slot* is the Layer 2 Ethernet interface

	Command or Action	Purpose

	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		that you want to configure as trusted or untrusted for DHCP snooping.
		
		• Enters interface configuration mode, where *port* / *slot* is the Layer 2 port-channel interface that you want to configure as trusted or untrusted for DHCP snooping.

Step 3	[**no**] **ip dhcp snooping trust** **Example:** `switch(config-if)# ip dhcp snooping trust`	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling the DHCP Relay Agent

You can enable or disable the DHCP relay agent. By default, the DHCP relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.

	Command or Action	Purpose
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Enabling or Disabling Option 82 for the DHCP Relay Agent

You can enable or disable the device to insert and remove Option 82 information on DHCP packets forwarded by the relay agent.

By default, the DHCP relay agent does not include Option 82 information in DHCP packets.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay information option**
	
	Example:
	`switch(config)# ip dhcp relay information`
	`option`
		Enables the DHCP relay agent to insert and remove Option 82 information on the packets that it forwards. The Option 82 information is in binary ifindex format by default. The **no** option disables this behavior.

Step 3	(Optional) [**no**] **ip dhcp relay information sub-option circuit-id format-type string**
	
	Example:
	`switch(config)# ip dhcp relay information`
	`sub-option circuit-id format-type string`
		Configures Option 82 to use encoded string format instead of the default binary ifindex format.

Step 4	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.

	Command or Action	Purpose
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring DHCP Server Addresses on an Interface

You can configure DHCP server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCP server IP addresses specified. The relay agent forwards replies from all DHCP servers to the host that sent the request.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP server is correctly configured.

Determine the IP address for each DHCP server that you want to configure on the interface.

If the DHCP server is in a different VRF instance than the interface, ensure that you have enabled VRF support.

> **Note** If an ingress router ACL is configured on an interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	
	   • **interface ethernet** *slot*/*port*[**.** *number*]
	   • **interface vlan** *vlan-id*
	   • **interface port-channel** *channel-id*[.*subchannel-id*]
	
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.
		
		• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you

	Command or Action	Purpose
		want to configure with a DHCP server IP address.
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCP server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.
Step 3	**ip dhcp relay address** *IP-address* **Example:** `switch(config-if)# ip dhcp relay address 10.132.7.120`	Configures an IP address for a DHCP server to which the relay agent forwards BOOTREQUEST packets received on this interface. To configure more than one IP address, use the **ip dhcp relay address** command once per address.
Step 4	(Optional) **show ip dhcp relay address** **Example:** `switch(config-if)# show ip dhcp relay address`	Displays all the configured DHCP server addresses.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a DHCP Static Binding

You can create a static DHCP source binding to a Layer 2 interface.

Before you begin

Ensure that you have enabled the DHCP snooping feature.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	**ip source binding** *IP-address MAC-address* **vlan** *vlan-id* {**interface ethernet** *slot/port* | **port-channel** *channel-no*} **Example:** switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3	Binds the static source address to the Layer 2 Ethernet interface.
Step 3	(Optional) **show ip dhcp snooping binding** **Example:** switch(config)# ip dhcp snooping binding	Shows the DHCP snooping static and dynamic bindings.
Step 4	(Optional) **show ip dhcp snooping binding dynamic** **Example:** switch(config)# ip dhcp snooping binding dynamic	Shows the DHCP snooping dynamic bindings.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Example

The following example shows how to create a static IP source entry associated with VLAN 100 on Ethernet interface 2/3:

```
switch# configure terminal
switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3
switch(config)#
```

Configuring DHCPv6 Relay Agent

Enabling or Disabling the DHCPv6 Relay Agent

You can enable or disable the DHCPv6 relay agent. By default, the DHCPv6 relay agent is disabled.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
	Example:	
	switch# configure terminal switch(config)#	
Step 2	[**no**] **ipv6 dhcp relay**	Enables the DHCPv6 relay agent. The **no** option disables the relay agent.
	Example:	
	switch(config)# ipv6 dhcp relay	
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*]	Displays the DHCPv6 relay configuration.
	Example:	
	switch(config)# show ipv6 dhcp relay	
Step 4	(Optional) **show running-config dhcp**	Displays the DHCP configuration.
	Example:	
	switch(config)# show running-config dhcp	
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.
	Example:	
	switch(config)# copy running-config startup-config	

Enabling or Disabling VRF Support for the DHCPv6 Relay Agent

You can configure the device to support the relaying of DHCPv6 requests that arrive on an interface in one VRF to a DHCPv6 server in a different VRF.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
	Example:	
	switch# configure terminal switch(config)#	

	Command or Action	Purpose
Step 2	**[no] ipv6 dhcp relay option vpn** **Example:** `switch(config)# ipv6 dhcp relay option vpn`	Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior.
Step 3	**[no] ipv6 dhcp relay option type cisco** **Example:** `switch(config)# ipv6 dhcp relay option type cisco`	Causes the DHCPv6 relay agent to insert virtual subnet selection (VSS) details as part of the vendor-specific option. The **no** option causes the DHCPv6 relay agent to insert VSS details as part of the VSS option (68), which is defined in RFC-6607. This command is useful when you want to use DHCPv6 servers that do not support RFC-6607 but allocate IPv6 addresses based on the client VRF name.
Step 4	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the DHCPv6 Relay Source Interface

You can configure the source interface for the DHCPv6 relay agent. By default, the DHCPv6 relay agent uses the relay agent address as the source address of the outgoing packet. Configuring the source interface enables you to use a more stable address (such as the loopback interface address) as the source address of relayed messages.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ipv6 dhcp relay source-interface** *interface*
	
	Example:
	
	`switch(config)# ipv6 dhcp relay`
	`source-interface loopback 2`
		Configures the source interface for the DHCPv6 relay agent.
		
		Note The DHCPv6 relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.

Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Verifying the DHCP Snooping Configuration

To display DHCP snooping configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the System Management Configuration Guide for your Cisco Nexus device.

Command	**Purpose**
show running-config dhcp	Displays the DHCP snooping configuration.
show ip dhcp relay	Displays the DHCP relay configuration.

Command	Purpose
show ipv6 dhcp relay [**interface** *interface*]	Displays the DHCPv6 relay global or interface-level configuration.
show ip dhcp snooping	Displays general information about DHCP snooping.

Displaying DHCP Bindings

Use the **show ip dhcp snooping binding** command to display the DHCP static and dynamic binding table. Use the **show ip dhcp snooping binding dynamic** to display the DHCP dynamic binding table.

For detailed information about the fields in the output from this command, see the *System Management Configuration Guide* for your Cisco Nexus device.

This example shows how to create a static DHCP binding and then verify the binding using the **show ip dhcp snooping binding** command.

```
switch# configuration terminal
switch(config)# ip source binding 10.20.30.40 0000.1111.2222 vlan 400 interface port-channel
 500

switch(config)# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type       VLAN  Interface
-----------------  ---------------  --------  ---------  ----  -------------
00:00:11:11:22:22  10.20.30.40      infinite  static     400   port-channel500
```

Clearing the DHCP Snooping Binding Database

You can remove entries from the DHCP snooping binding database, including a single entry, all entries associated with an interface, or all entries in the database.

Before you begin

Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **clear ip dhcp snooping binding** **Example:** `switch# clear ip dhcp snooping binding`	Clears all entries from the DHCP snooping binding database.
Step 2	(Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*] **Example:**	Clears entries associated with a specific Ethernet interface from the DHCP snooping binding database.

	Command or Action	Purpose
	`switch# clear ip dhcp snooping binding interface ethernet 1/4`	
Step 3	(Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*] **Example:** `switch# clear ip dhcp snooping binding interface port-channel 72`	Clears entries associated with a specific port-channel interface from the DHCP snooping binding database.
Step 4	(Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] } **Example:** `switch# clear ip dhcp snooping binding vlan 23 mac 0060.3aeb.54f0 ip 10.34.54.9 interface ethernet 2/11`	Clears a single, specific entry from the DHCP snooping binding database.
Step 5	(Optional) **show ip dhcp snooping binding** **Example:** `switch# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Clearing DHCP Relay Statistics

Use the **clear ip dhcp relay statistics** command to clear the global DHCP relay statistics.

Use the **clear ip dhcp relay statistics interface** *interface* command to clear the DHCP relay statistics for a particular interface.

Use the **clear ip dhcp relay statistics interface** *interface* **serverip** *ip-address* [**use-vrf** *vrf-name*] command to clear the DHCP relay statistics at the server level for a particular interface.

Clearing DHCPv6 Relay Statistics

Use the **clear ipv6 dhcp relay statistics** command to clear the global DHCPv6 relay statistics.

Use the **clear ipv6 dhcp relay statistics interface** *interface* command to clear the DHCPv6 relay statistics for a particular interface.

Use the **clear ipv6 dhcp relay statistics interface** *interface* **server-ip** *ip-address* [**use-vrf** *vrf-name*] command to clear the DHCPv6 relay statistics at the server level for a particular interface.

Monitoring DHCP

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.

Use the **show ip dhcp relay statistics** [**interface** *interface* [**serverip** *ip-address* [**use-vrf** *vrf-name*]]] command to monitor DHCP relay statistics at the global, server, or interface level.

Use the (Optional) **show ip dhcp snooping statistics vlan** [*vlan-id*] **interface** [**ethernet**|*port-channel*][*id*] command to know the exact statistics about snooping statistics per interface under a vlan.

Configuration Examples for DHCP Snooping

The following example shows how to enable DHCP snooping on two VLANs, with Option 82 support enabled and Ethernet interface 2/5 trusted because the DHCP server is connected to that interface:

```
feature dhcp
ip dhcp snooping
ip dhcp snooping info option

interface Ethernet 2/5
  ip dhcp snooping trust
ip dhcp snooping vlan 1
ip dhcp snooping vlan 50
```

Configuring IPv6 First-Hop Security

This chapter describes how to configure IPv6 First-Hop Security on Cisco NX-OS devices and includes the following sections:

Introduction to First-Hop Security

The Layer 2 and Layer 3 switches operate in the Layer 2 domains with technologies, such as server virtualization, Overlay Transport Virtualization (OTV), and Layer 2 mobility. These devices are sometimes referred to as "first hops", specifically when they are facing end nodes. The First-Hop Security feature provides end node protection and optimizes link operations on IPv6 or dual-stack networks.

First-Hop Security (FHS) is a set of features to optimize IPv6 link operation, as well as help with scale in large L2 domains. These features provide protection from a wide host of rogue or mis-configured users, and this can be extended with additional features for different deployment scenarios, or attack vectors.

The following FHS features are supported:

- IPv6 RA Guard
- DHCPv6 Guard
- IPv6 Snooping

Note Use the **feature dhcp** command to enable the FHS feature.

IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 snooping, DHCPv6 guard, and IPv6 RA guard are IPv6 global policies features. Every time IPv6 snooping, DHCPv6 guard, or RA guard is configured globally, the policy attributes are stored in the software policy database. The policy is then applied to an interface or VLAN, and the software policy database entry is updated to include this interface to which the policy is applied.

All port level FHS policies are programmed in the ifacl region, while the VLAN level policies are programmed in the FHS region. Use the **hardware profile tcam region fhs** *tcam_size* command to configure the FHS region. The range for the TCAM size is 0-4096.

All FHS packets take the **copp-s-dhcpreq** queue for software processing.

IPv6 First-Hop Security Binding Table

A database of table of IPv6 neighbors connected to the device is created from information sources, such as IPv6 snooping. This database, or binding, table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

RA Guard

Overview of IPv6 RA Guard

The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The IPv6 RA Guard feature analyzes these RAs and filters out RAs that are sent by unauthorized devices. In host mode, all RA and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 (L2) device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

Guidelines and Limitations of IPv6 RA Guard

The guidelines and limitations of IPv6 RA Guard are as follows:

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.

- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.

- This feature can be configured on a switch port interface in the ingress direction.

- This feature supports host mode and router mode.

- This feature is supported only in the ingress direction; it is not supported in the egress direction.

- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.

- Packets dropped by the IPv6 RA Guard feature can be spanned.

- IPv6 RA Guard cannot be enabled if SFLOW is enabled.

- IPv6 RA Guard cannot be enabled on VXLAN ports.

- If the **platform ipv6 acl icmp optimize neighbor-discovery** command is configured, the IPv6 RA Guard feature cannot be configured and an error message will be displayed. This command adds default global Internet Control Message Protocol (ICMP) entries that will override the RA guard ICMP entries.

DHCPv6 Guard

Overview of DHCP—DHCPv6 Guard

The DHCPv6 Guard feature blocks DHCP reply and advertisement messages that originate from unauthorized DHCP servers and relay agents that forward DHCP packets from servers to clients. Client messages or messages sent by relay agents from clients to servers are not blocked. The filtering decision is determined by the device role assigned to the receiving switch port, trunk, or VLAN.

Packets are classified into one of the three DHCP type messages. All client messages are always switched regardless of device role. DHCP server messages are only processed further if the device role is set to server. Further processing of DHCP server advertisements occurs for server preference checking.

If the device is configured as a DHCP server, all the messages need to be switched, regardless of the device role configuration.

Guidelines and Limitations of DHCPv6 Guard

The guidelines and limitations of DHCPv6 Guard are as follows:

- If a packet arriving from the DHCP server is a Relay Forward or a Relay Reply, only the device role is checked. In addition, IPv6 DHCP Guard does not apply the policy for a packet sent out by the local relay agent running on the switch.

- DHCP Guard cannot be enabled if SFLOW is enabled.

- DHCP Guard is not supported on VXLAN ports.

IPv6 Snooping

Overview of IPv6 Snooping

IPv6 "snooping," feature bundles several Layer 2 IPv6 first-hop security features, which operates at Layer 2, or between Layer 2 and Layer 3, and provides IPv6 features with security and scalability. This feature mitigates some of the inherent vulnerabilities for the neighbor discovery mechanism, such as attacks on duplicate address detection (DAD), address resolution, device discovery, and the neighbor cache.

IPv6 snooping learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables and analyzes snooping messages in order to build a trusted binding table. IPv6 snooping messages that do not have valid bindings are dropped. An IPv6 snooping message is considered trustworthy if its IPv6-to-MAC mapping is verifiable.

When IPv6 snooping is configured on a target (which varies depending on platform target support and may include device ports, switch ports, Layer 2 interfaces, Layer 3 interfaces, and VLANs), capture instructions are downloaded to the hardware to redirect the snooping protocol and Dynamic Host Configuration Protocol (DHCP) for IPv6 traffic up to the switch integrated security features (SISF) infrastructure in the routing device. For snooping traffic, Neighbor Discovery Protocol (NDP) messages are directed to SISF. For DHCPv6, UDP messages sourced from dhcvp6_client and dhcvp_server ports are redirected.

IPv6 snooping registers its "capture rules" to the classifier, which aggregates all rules from all features on a given target and installs the corresponding ACL down into the platform-dependent modules. Upon receiving redirected traffic, the classifier calls all entry points from any registered feature (for the target on which the traffic is being received), including the IPv6 snooping entry point. This entry point is the last to be called, so any decision (such as drop) made by another feature supersedes the IPv6 snooping decision.

IPv6 snooping provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

Additionally, IPv6 snooping is the foundation for many other IPv6 features that depend on an accurate binding table. It inspects snooping and DHCP messages on a link to glean addresses, and then populates the binding table with these addresses. This feature also enforces address ownership and limits the number of addresses any given node is allowed to claim.

Guidelines and Limitations for IPv6 Snooping

The guidelines and limitations of IPv6 Snooping are as follows:

- You must perform the same configurations on both the vPC peers.

- The IPv6 Snooping feature is supported only in the hardware when the ternary content addressable memory (TCAM) is programmed.

- The IPv6 Snooping feature can be configured on a switch port interface or on the VLAN only ingress port.

- The tracking functionality of the IPv6 snooping policy will not work if the Neighbor Discovery protocol is disabled in the configured IPv6 snooping policy.

- For the IPv6 Snooping to learn DHCP bindings, it must see both, the server and the client replies. A IPv6 snooping policy must be attached to both the client facing interface (or VLAN), as well as the DHCP server facing interface (or VLAN). In the case of a DHCP Relay, a IPv6 snooping policy must be attached at the VLAN level to see the server replies.

How to Configure IPv6 FHS

Configuring the IPv6 RA Guard Policy on the Device

> **Note** When the **ipv6 nd raguard** command is configured on ports, router solicitation messages are not replicated to these ports. To replicate router solicitation messages, all ports that face routers must be set to the router role.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy policy1`	Defines the RA guard policy name and enters RA guard policy configuration mode.
Step 3	**device-role** {**host** | **router** | **monitor** | **switch**} **Example:** `Device(config-ra-guard)# device-role router`	Specifies the role of the device attached to the port.
Step 4	**hop-limit** {**maximum** | **minimum** *limit*} **Example:** `Device(config-ra-guard)# hop-limit minimum 3`	(Optional) Enables verification of the advertised hop count limit. • If not configured, this check will be bypassed.
Step 5	**managed-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# managed-config-flag on`	(Optional) Enables verification that the advertised managed address configuration flag is on. • If not configured, this check will be bypassed.
Step 6	**other-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# other-config-flag on`	(Optional) Enables verification of the advertised "other" configuration parameter.

	Command or Action	Purpose
Step 7	**router-preference maximum** {**high**	**low**	**medium**} **Example:** Device(config-ra-guard)# router-preference maximum high	(Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit.
Step 8	**trusted-port** **Example:** Device(config-ra-guard)# trusted-port	(Optional) Specifies that this policy is being applied to trusted ports. • All RA guard policing will be disabled.
Step 9	**exit** **Example:** Device(config-ra-guard)# exit	Exits RA guard policy configuration mode and returns to global configuration mode.

Configuring IPv6 RA Guard on an Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 2	**interface** *type* *number* **Example:** Device(config)# interface fastethernet 3/13	Specifies an interface type and number, and places the device in interface configuration mode.
Step 3	**ipv6 nd raguard attach-policy** [*policy-name*] **Example:** Device(config-if)# ipv6 nd raguard attach-policy	Applies the IPv6 RA Guard feature to a specified interface.
Step 4	**exit** **Example:** Device(config-if)# exit	Exits interface configuration mode.
Step 5	**show ipv6 nd raguard policy** [*policy-name*] **Example:** switch# show ipv6 nd raguard policy host Policy host configuration: device-role host	Displays the RA guard policy on all interfaces configured with the RA guard.

	Command or Action	Purpose
| | ```
Policy applied on the following
interfaces:

  Et0/0        vlan all
  Et1/0        vlan all
``` | |
| Step 6 | **debug ipv6 snooping raguard** [*filter* \| *interface* \| *vlanid*]<br><br>**Example:**<br><br>```
Device# debug ipv6 snooping raguard
``` | Enables debugging for IPv6 RA guard snooping information. |

Configuring DHCP—DHCPv6 Guard

Procedure

	Command or Action	Purpose
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>```
Device# configure terminal
``` | Enters global configuration mode. |
| Step 2 | **ipv6 dhcp guard policy** *policy-name*<br><br>**Example:**<br><br>```
Device(config)# ipv6 dhcp guard policy
 pol1
``` | Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode. |
| Step 3 | **device-role** {**client** \| **server**}<br><br>**Example:**<br><br>```
Device(config-dhcp-guard)# device-role
 server
``` | Specifies the device role of the device attached to the target (interface or VLAN). |
| Step 4 | **preference min** *limit*<br><br>**Example:**<br><br>```
Device(config-dhcp-guard)# preference
min 0
``` | (Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed. |
| Step 5 | **preference max** *limit*<br><br>**Example:**<br><br>```
Device(config-dhcp-guard)# preference
max 255
``` | (Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed. |

	Command or Action	**Purpose**
Step 6	**trusted-port** **Example:** Device(config-dhcp-guard)# trusted-port	(Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled.
Step 7	**exit** **Example:** Device(config-dhcp-guard)# exit	Exits DHCP guard configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** Device(config)# interface GigabitEthernet 0/2/0	Specifies an interface and enters interface configuration mode.
Step 9	**switchport** **Example:** Device(config-if)# switchport	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 10	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** Device(config-if)# ipv6 dhcp guard attach-policy pol1	Attaches a DHCPv6 guard policy to an interface.
Step 11	**exit** **Example:** Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 12	**vlan configuration** *vlan-id* **Example:** Device(config)# vlan configuration 1	Specifies a VLAN and enters VLAN configuration mode.
Step 13	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** Device(config-vlan-config)# ipv6 dhcp guard attach-policy pol1	Attaches a DHCPv6 guard policy to a VLAN.
Step 14	**exit** **Example:** Device(config-vlan-config)# exit	Exits VLAN configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 15	**exit** **Example:** Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 16	**show ipv6 dhcp guard policy** [*policy-name*] **Example:** Device# show ipv6 dhcp policy guard pol1	(Optional) Displays the policy configuration as well as all the interfaces where the policy is applied.

Configuring IPv6 Snooping

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 2	**ipv6 snooping policy** *policy-name* **Example:** Device(config)# ipv6 snooping policy policy1	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.
Step 3	**device-role** { **node** | **switch** } **Example:** Device(config-snoop-policy)# device-node switch	Specifies the role of the device attached to the target (interface or VLAN): • node—Is the default. Bindings are created and entries are probed. • switch—Entries are not probed and when a trusted port is enabled, bindings are not created.
Step 4	**[no] limit address-count** **Example:** Device(config-snoop-policy)# limit address-count 500	Limits the number of binding entries, a no limit address-count means no limit.
Step 5	**[no] protocol** *dhcp* | *ndp* **Example:** Device(config-snoop-policy)# protocol dhcp Device(config-snoop-policy)# protocol ndp	Turns on or switches off either DHCP or NDP gleaning.

	Command or Action	**Purpose**
Step 6	**trusted-port** **Example:** Device(config-snoop-policy)# trusted-port	Specifies that the policy be applied to a trusted port. If an entry is a trusted-port, none of it's traffic will be blocked or dropped.
Step 7	**security-level** *glean* | *guard* | *inspect* **Example:** Device(config-snoop-policy)# security-level guard	Specifies the type of security level applied to the policy, such as: • glean—learns bindings but does not drop packets. • inspect—learns bindings and drops packets in case it detects an issue, such as an address theft. • guard—works like inspect, but in addition drops IPv6, ND, RA, and IPv6 DHCP server packets in case of a threat.
Step 8	**tracking** **Example:** Device(config-snoop-policy)# tracking enable	Enables tracking.
Step 9	**exit** **Example:** Device(config-snoop-policy)# exit	Exits snooping configuration mode and returns to global configuration mode.
Step 10	**interface** *type-number* **Example:** Device(config-if)# interface ethernet 1/25	Specifies an interface and enters interface configuration mode.
Step 11	**[no] switchport** **Example:** Device(config-if)# switchport	Switches between Layer 2 and Layer 3 mode.
Step 12	**ipv6 snooping attach-policy** *policy-name* **Example:** Device(config-if)# ipv6 snooping attach-policy policy1	Attaches the IPv6 snooping policy to an interface.
Step 13	**exit** **Example:** Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 14	**vlan configuration** *vlan-id* **Example:**	Specifies a VLAN and enters VLAN configuration mode.

	Command or Action	Purpose
	`Device(config)# vlan configuration 333`	
Step 15	**ipv6 snooping attach-policy** *policy-name* **Example:** `Device(config-vlan-config)# ipv6 snooping attach-policy policy1`	Attaches the IPv6 snooping policy to a VLAN.
Step 16	**exit** **Example:** `Device(config-vlan-config)# exit`	Exits VLAN configuration mode and returns to global configuration mode.
Step 17	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 18	**show ipv6 snooping policy** *policy-name* **Example:** `Device(config)# show ipv6 snooping policy policy1`	Displays the policy configuration and the interfaces where the policy is applied.

Configuring IPv6 First-Hop Security Binding Table

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 neighbor binding vlan** *vlan-id* {**interface** *type number* | *ipv6-address* | *mac-address*} [**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*] **Example:** `Device(config)# ipv6 neighbor binding vlan 100 interface Ethernet 0/0 reachable-lifetime 100`	Adds a static entry to the binding table database.
Step 3	**ipv6 neighbor binding max-entries** *entries* [**vlan-limit** *number* | **interface-limit** *number* | **mac-limit** *number*] **Example:**	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.

	Command or Action	**Purpose**
	`Device(config)# ipv6 neighbor binding max-entries 100`	
Step 4	**ipv6 neighbor binding logging** **Example:** `Device(config)# ipv6 neighbor binding logging`	Enables the logging of binding table main events.
Step 5	**ipv6 neighbor tracking retry-interval** *value* **Example:** `Device(config)# ipv6 neighbor binding retry-interval 8`	Tracks entries in the binding table.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**show ipv6 neighbor binding** [**vlan** *vlan-id* | **interface** *type number* | **ipv6** *ipv6-address* | **mac** *mac-address*] **Example:** `Device# show ipv6 neighbor binding`	Displays the contents of a binding table.

Verifying and Troubleshooting IPv6 Snooping

Procedure

	Command or Action	**Purpose**
Step 1	**show ipv6 snooping capture-policy** [**interface** *type number*] **Example:** `Device# show ipv6 snooping capture-policy interface ethernet 0/0`	Displays snooping message capture policies.
Step 2	**show ipv6 snooping counter** [**interface** *type number*] **Example:** `Device# show ipv6 snooping counter interface FastEthernet 4/12`	Displays information about the packets counted by the interface counter.

	Command or Action	**Purpose**
Step 3	**show ipv6 snooping features** **Example:** `Device# show ipv6 snooping features`	Displays information about snooping features configured on the device.
Step 4	**show ipv6 snooping policies** [**interface** *type number*] **Example:** `Device# show ipv6 snooping policies`	Displays information about the configured policies and the interfaces to which they are attached.
Step 5	**debug ipv6 snooping** **Example:** `Device# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

Configuration Examples

Example: IPv6 RA Guard Configuration

```
switch(config)# interface fastethernet 3/13

switch(config-if)# ipv6 nd raguard attach-policy

Device# show running-config interface fastethernet 3/13

Building configuration...
Current configuration : 129 bytes
!
interface FastEthernet3/13
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

Example: Configuring DHCP—DHCPv6 Guard

The following example displays a sample configuration for DHCPv6 Guard:

```
configure terminal
ipv6 dhcp guard policy pol1
 device-role server
 preference min 0
 preference max 255
 trusted-port
```

```
interface GigabitEthernet 0/2/0
 switchport
 ipv6 dhcp guard attach-policy pol1
 vlan configuration 1
  ipv6 dhcp guard attach-policy pol1
show ipv6 dhcp guard policy pol1
```

Example: Configuring IPv6 First-Hop Security Binding Table

```
config terminal
 ipv6 neighbor binding vlan 100 2001:db8::1 interface ethernet3/0
 ipv6 neighbor binding max-entries 100
 ipv6 neighbor binding logging
 ipv6 neighbor binding retry-interval 8
 exit
show ipv6 neighbor binding
```

Example: Configuring IPv6 Snooping

```
switch (config)# ipv6 snooping policy policy1
switch(config-ipv6-snooping)# ipv6 snooping attach-policy policy1
switch(config-ipv6-snooping)# exit
.
.
.
switch# show ipv6 snooping policies policy1
Policy policy1 configuration:
  trusted-port
  device-role node
Policy applied on the following interfaces:
   Et0/0       vlan all
   Et1/0       vlan all
Policy applied on the following vlans:
   vlan 1-100,200,300-400
```

Additional References for IPv6 First-Hop Security

This section includes additional information related to configuring IPv6 First-Hop Security.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference*

Configuring Dynamic ARP Inspection

This chapter contains the following sections:

Information About DAI

ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

ARP Spoofing Attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your Layer 2 network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic intended for other hosts on the subnet.

Figure 5: ARP Cache Poisoning

This figure shows an example of ARP cache poisoning.

Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address associated with IP address IB. When the device and host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds, the device and host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the device, host A, and host B by broadcasting two forged ARP responses with bindings: one for a host with an IP address of IA and a MAC address of MC and another for a host with the IP address of IB and a MAC address of MC. Host B and the device then use the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Likewise, host A and the device use the MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a *man-in-the middle* attack.

DAI and ARP Spoofing Attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco Nexus device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports

- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination

- Drops invalid ARP packets

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the device. It can also contain static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header.

Interface Trust States and Network Security

DAI associates a trust state with each interface on the device. Packets that arrive on trusted interfaces bypass all DAI validation checks, and packets that arrive on untrusted interfaces go through the DAI validation process.

In a typical network configuration, the guidelines for configuring the trust state of interfaces are as follows:

Untrusted

Interfaces that are connected to hosts

Trusted

Interfaces that are connected to devices

With this configuration, all ARP packets that enter the network from a device bypass the security check. No other validation is needed at any other place in the VLAN or in the network.

> ⚠️
>
> **Caution** Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

Figure 6: ARP Packet Validation on a VLAN Enabled for DAI

The following figure, assume that both device A and device B are running DAI on the VLAN that includes host 1 and host 2. If host 1 and host 2 acquire their IP addresses from the DHCP server connected to device A, only device A binds the IP-to-MAC address of host 1. If the interface between device A and device B is untrusted, the ARP packets from host 1 are dropped by device B and connectivity between host 1 and host 2 is lost.

If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though device B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a device that runs DAI do not poison the ARP caches of other hosts in the network; however, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a device that runs DAI.

If some devices in a VLAN run DAI and other devices do not, the guidelines for configuring the trust state of interfaces on a device that runs DAI becomes the following:

Untrusted

Interfaces that are connected to hosts or to devices that *are not* running DAI

Trusted

Interfaces that are connected to devices that *are* running DAI

To validate the bindings of packets from devices that do not run DAI, configure ARP ACLs on the device that runs DAI. When you cannot determine the bindings, isolate at Layer 3 the devices that run DAI from devices that do not run DAI.

Note Depending on your network setup, you may not be able to validate a given ARP packet on all devices in the VLAN.

Logging DAI Packets

Cisco NX-OS maintains a buffer of log entries about DAI packets processed. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You can also specify the type of packets that are logged. By default, aCisco Nexus device logs only packets that DAI drops.

If the log buffer overflows, the device overwrites the oldest DAI log entries with newer entries. You can configure the maximum number of entries in the buffer.

Note Cisco NX-OS does not generate system messages about DAI packets that are logged.

Licensing Requirements for DAI

This table shows the licensing requirements for DAI.

Product	License Requirement
Cisco NX-OS	DAI requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

Prerequisites for DAI

• You must enable the DHCP feature before you can configure DAI.

Guidelines and Limitations for DAI

DAI has the following configuration guidelines and limitations:

- DAI is an ingress security feature; it does not perform any egress checking.

- DAI is not effective for hosts connected to devices that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, you should separate the domain with DAI from domains without DAI. This separation secures the ARP caches of hosts in the domain with DAI.

- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, DHCP snooping needs only to be enabled. If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, you must configure DHCP snooping on the same VLANs on which you configure DAI.

- When you use the **feature dhcp** command to enable the DHCP feature, there is a delay of approximately 30 seconds before the I/O modules receive the DHCP or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with the DHCP feature disabled to a configuration with the DHCP feature enabled. For example, if you use the Rollback feature to revert to a configuration that enables the DHCP feature, the I/O modules receive the DHCP and DAI configuration approximately 30 seconds after you complete the rollback.

- DAI is supported on access ports, trunk ports, port-channel ports, and private VLAN ports.

- The DAI trust configuration of a port channel determines the trust state of all physical ports that you assign to the port channel. For example, if you have configured a physical port as a trusted interface and then you add that physical port to a port channel that is an untrusted interface, the physical port becomes untrusted.

- When you remove a physical port from a port channel, the physical port does not retain the DAI trust state configuration of the port channel.

- When you change the trust state on the port channel, the device configures a new trust state on all the physical ports that comprise the channel.

- If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is enabled and that you have configured the static IP-MAC address bindings.

- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is enabled.

Default Settings for DAI

This table lists the default settings for DAI parameters.

Table 18: Default DAI Parameters

Parameters	Default
DAI	Disabled on all VLANs.

Parameters	Default
Interface trust state	All interfaces are untrusted.
Validation checks	No checks are performed.
Log buffer	When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

Configuring DAI

Enabling or Disabling DAI on VLANs

You can enable or disable DAI on VLANs. By default, DAI is disabled on all VLANs.

Before you begin

If you are enabling DAI, ensure the following:

- Ensure that the DHCP feature is enabled.

- The VLANs on which you want to enable DAI are configured.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	 Example:
	 ```switch# configure terminal```
	 ```switch(config)#```
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection vlan** *list*
	
	 Example:
	 ```switch(config)# ip arp inspection vlan```
	 ```13```
		Enables DAI for the specified list of VLANs. The **no** option disables DAI for the specified VLANs.

Step 3	(Optional) **show ip arp inspection vlan** *list*
	
	 Example:
	 ```switch(config)# show ip arp inspection```
	 ```vlan 13```
		Shows the DAI status for the specified list of VLANs.

	Command or Action	Purpose

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring the DAI Trust State of a Layer 2 Interface

You can configure the DAI interface trust state of a Layer 2 interface. By default, all interfaces are untrusted.

A device forwards ARP packets that it receives on a trusted Layer 2 interface but does not check them.

On untrusted interfaces, the device intercepts all ARP requests and responses and verifies that the intercepted packets have valid IP-MAC address bindings before updating the local cache and forwarding the packet to the appropriate destination. If the device determines that packets have invalid bindings, it drops the packets and logs them according to the logging configuration.

Before you begin

If you are enabling DAI, ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number* **/** *slot*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **ip arp inspection trust**
	
	Example:
	`switch(config-if)# ip arp inspection`
	`trust`
		Configures the interface as a trusted ARP interface. The **no** option configures the interface as an untrusted ARP interface.

Step 4	(Optional) **show ip arp inspection interface** *type number* **/** *slot*
	
	Example:
	`switch(config-if)# show ip arp inspection`
	`interface ethernet 2/1`
		Displays the trust state and the ARP packet rate for the specified interface.

	Command or Action	Purpose
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Additional Validation

You can enable or disable additional validation of ARP packets. By default, no additional validation of ARP packets is enabled. When no additional validation is configured, the source MAC address and the source IP address check against the IP-to-MAC binding entry for ARP packets are done by using the Ethernet source MAC address (not the ARP sender MAC address) and the ARP sender IP address.

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can use the following keywords with the **ip arp inspection validate** command to implement additional validations:

dst-mac

Checks the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

ip

Checks the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

src-mac

Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, follow these guidelines:

- You must specify at least one of the keywords. You can specify one, two, or all three keywords.

- Each **ip arp inspection validate** command that you enter replaces the configuration from any previous commands. If you enter an **ip arp inspection validate** command to enable src-mac and dst-mac validations, and a second **ip arp inspection validate** command to enable ip validation, the src-mac and dst-mac validations are disabled when you enter the second command.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	**Purpose**
	`switch# configure terminal` `switch(config)#`	

Step 2	[**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}
	Example:
	`switch(config)# ip arp inspection`
	`validate src-mac dst-mac ip`
		Enables additional DAI validation, or if you use the **no** option, disables additional DAI validation.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.

Step 4	(Optional) **copy running-config startup-config**
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring the DAI Logging Buffer Size

You can configure the DAI logging buffer size. The default buffer size is 32 messages.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection log-buffer entries** *number*
	Example:
	`switch(config)# ip arp inspection`
	`log-buffer entries 64`
		Configures the DAI logging buffer size. The **no** option reverts to the default buffer size, which is 32 messages. The buffer size can be between 1 and 1024 messages.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.

Step 4	(Optional) **copy running-config startup-config**
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring DAI Log Filtering

You can configure how the device determines whether to log a DAI packet. By default, the device logs DAI packets that are dropped.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: • **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings all** • **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings none** • **ip arp inspection vlan** *vlan-list* **logging dhcp-bindingspermit** • **no ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**} **Example:** `switch(config)# ip arp inspection vlan 100 dhcp-bindings permit`	Configures DAI log filtering, as follows. The **no** option removes DAI log filtering. • Logs all packets that match DHCP bindings. • Does not log packets that match DHCP bindings. • Logs packets permitted by DHCP bindings. • Removes DAI log filtering.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the DAI Configuration

To display the DAI configuration information, perform one of the following tasks.

Command	Purpose
show ip arp inspection	Displays the status of DAI.
show ip arp inspection interface ethernet	Displays the trust state.

Command	Purpose
show ip arp inspection vlan	Displays the DAI configuration for a specific VLAN.
show arp access-lists	Displays ARP ACLs.
show ip arp inspection log	Displays the DAI log configuration.

Monitoring and Clearing DAI Statistics

Configuration Examples for DAI

Example 1-Two Devices Support DAI

These procedures show how to configure DAI when two devices support DAI.

Figure 7: Two Devices Supporting DAI

The following figure shows the network configuration for this example. Host 1 is connected to device A, and Host 2 is connected to device B. Both devices are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to device A. Both hosts acquire their IP addresses from the same DHCP server. Device A has the bindings for Host 1 and Host 2, and device B has the binding for Host 2. Device A Ethernet interface 2/3 is connected to the device B Ethernet interface 1/4.

DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically-assigned IP addresses.

- This configuration does not work if the DHCP server is moved from device A to a different location.

- To ensure that this configuration does not compromise security, configure Ethernet interface 2/3 on device A and Ethernet interface 1/4 on device B as trusted.

Configuring Device A

To enable DAI and configure Ethernet interface 2/3 on device A as trusted, follow these steps:

Procedure

Step 1   While logged into device A, verify the connection between device A and device B.

```
switchA# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID            Local Intrfce  Hldtme  Capability  Platform     Port ID
switchB              Ethernet2/3    177     R S I    WS-C2960-24TC Ethernet1/4
switchA#
```

Step 2   Enable DAI on VLAN 1 and verify the configuration.

```
switchA# config t
switchA(config)# ip arp inspection vlan 1
switchA(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchA(config)#
```

Step 3   Configure Ethernet interface 2/3 as trusted.

```
switchA(config)# interface ethernet 2/3
switchA(config-if)# ip arp inspection trust
switchA(config-if)# exit
switchA(config)# exit
switchA# show ip arp inspection interface ethernet 2/3
 Interface      Trust State   Rate (pps)   Burst Interval
 -------------  -----------   ----------   --------------
 Ethernet2/3    Trusted          15            5
```

Step 4   Verify the bindings.

```
switchA# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type          VLAN  Interface
-----------------  ---------------  --------  ------------  ----  -------------
00:60:0b:00:12:89  10.0.0.1         0         dhcp-snooping  1     Ethernet2/3
switchA#
```

Step 5   Check the statistics before and after DAI processes any packets.

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
```

```
DMAC Fails-ARP Res = 0
IP Fails-ARP Req  = 0
IP Fails-ARP Res  = 0
switchA#
```

If host 1 sends out two ARP requests with an IP address of 10.0.0.1 and a MAC address of 0002.0002.0002, both requests are permitted, and are shown as follows:

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
```

If host 1 tries to send an ARP request with an IP address of 10.0.0.3, the packet is dropped and an error message is logged.

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Ethernet2/3, vlan
1.([0002.0002.0002/10.0.0.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Fri Jul 13 2008])
```

The statistics display as follows:

```
switchA# show ip arp inspection statistics vlan 1
switchA#
Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 2
ARP Res Dropped    = 0
DHCP Drops         = 2
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

Configuring Device B

To enable DAI and configure Ethernet interface 1/4 on device B as trusted, follow these steps:

Procedure

Step 1   While logged into device B, verify the connection between device B and device A.

```
switchB# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID          Local Intrfce  Hldtme  Capability  Platform      Port ID
switchA            Ethernet1/4    120     R S I    WS-C2960-24TC Ethernet2/3
switchB#
```

Step 2   Enable DAI on VLAN 1, and verify the configuration.

```
switchB# config t
switchB(config)# ip arp inspection vlan 1
switchB(config)# show ip arp inspection vlan 1
Source Mac Validation     : Disabled
Destination Mac Validation : Disabled
IP Address Validation     : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchB(config)#
```

Step 3   Configure Ethernet interface 1/4 as trusted.

```
switchB(config)# interface ethernet 1/4
switchB(config-if)# ip arp inspection trust
switchB(config-if)# exit
switchB(config)# exit
switchB# show ip arp inspection interface ethernet 1/4
 Interface        Trust State   Rate (pps)    Burst Interval
 -------------    -----------   ----------    --------------
 Ethernet1/4      Trusted           15            5
switchB#
```

Step 4   Verify the list of DHCP snooping bindings.

```
switchB# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type          VLAN  Interface
-----------------  ---------------  --------  ------------  ----  -------------
00:01:00:01:00:01  10.0.0.2         4995      dhcp-snooping  1    Ethernet1/4
switchB#
```

Step 5   Check the statistics before and after DAI processes any packets.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 sends out an ARP request with the IP address 10.0.0.2 and the MAC address 0001.0001.0001, the packet is forwarded and the statistics are updated.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 attempts to send an ARP request with the IP address 10.0.0.1, DAI drops the request and logs the following system message:

```
00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Ethernet1/4, vlan
1.([0001.0001.0001/10.0.0.1/0000.0000.0000/0.0.0.0/01:53:21 UTC Fri Jun 13 2008])
```

The statistics display as follows:

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 1
ARP Res Dropped    = 0
DHCP Drops         = 1
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

Configuring 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on Cisco NX-OS devices and includes the following sections:

About 802.1X

802.1X defines a client-server based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a Cisco NX-OS device port.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles.

Figure 8: 802.1X Device Roles

This figure shows the device roles in 802.1X.

The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and Cisco NX-OS device services and responds to requests from the Cisco NX-OS device. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating device.

> **Note** To resolve Windows XP network connectivity and Cisco 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL:
> http://support.microsoft.com/support/kb/articles/Q303/5/97.ASP

Authentication server

The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the Cisco NX-OS device regarding whether the supplicant is authorized to access the LAN and Cisco NX-OS device services. Because the Cisco NX-OS device acts as the proxy, the authentication service is transparent to the supplicant. The Remote Authentication Dial-In User Service (RADIUS) security device with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server, version 3.0. RADIUS uses a supplicant-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

Authenticator

The authenticator controls the physical access to the network based on the authentication status of the supplicant. The authenticator acts as an intermediary (proxy) between the supplicant and the authentication server, requesting identity information from the supplicant, verifying the requested identity information with the authentication server, and relaying a response to the supplicant. The authenticator includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the authenticator receives EAPOL frames and relays them to the authentication server, the authenticator strips off the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format. This encapsulation process does not modify or examine the EAP frames, and the authentication server must support EAP within the native frame format. When the authenticator receives frames from the authentication server, the authenticator removes the server's frame header, leaving the EAP frame, which the authenticator then encapsulates for Ethernet and sends to the supplicant.

> **Note** The Cisco NX-OS device can only be an 802.1X authenticator.

Authentication Initiation and Message Exchange

Either the authenticator (Cisco NX-OS device) or the supplicant (client) can initiate authentication. If you enable authentication on a port, the authenticator must initiate authentication when it determines that the port link state transitions from down to up. The authenticator then sends an EAP-request/identity frame to the supplicant to request its identity (typically, the authenticator sends an initial identity/request frame followed by one or more requests for authentication information). When the supplicant receives the frame, it responds with an EAP-response/identity frame.

If the supplicant does not receive an EAP-request/identity frame from the authenticator during bootup, the supplicant can initiate authentication by sending an EAPOL-start frame, which prompts the authenticator to request the supplicant's identity.

Note   If 802.1X is not enabled or supported on the network access device, the Cisco NX-OS device drops any EAPOL frames from the supplicant. If the supplicant does not receive an EAP-request/identity frame after three attempts to start authentication, the supplicant transmits data as if the port is in the authorized state. A port in the authorized state means that the supplicant has been successfully authenticated.

When the supplicant supplies its identity, the authenticator begins its role as the intermediary, passing EAP frames between the supplicant and the authentication server until authentication succeeds or fails. If the authentication succeeds, the authenticator port becomes authorized.

The specific exchange of EAP frames depends on the authentication method being used.

Figure 9: Message Exchange

This figure shows a message exchange initiated by the supplicant using the One-Time-Password (OTP) authentication method with a RADIUS server. The OTP authentication device uses a secret pass-phrase to generate a sequence of one-time (single use) passwords.

The secret pass-phrase of the user never crosses the network at any time such as during authentication or during pass-phrase changes.

Authenticator PAE Status for Interfaces

When you enable 802.1X on an interface, the Cisco NX-OS software creates an authenticator port access entity (PAE) instance. An authenticator PAE is a protocol entity that supports authentication on the interface. When you disable 802.1X on the interface, the Cisco NX-OS software does not automatically clear the authenticator PAE instances. You can explicitly remove the authenticator PAE from the interface and then reapply it, as needed.

Ports in Authorized and Unauthorized States

The authenticator port state determines if the supplicant is granted access to the network. The port starts in the unauthorized state. In this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a supplicant is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the supplicant to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the authenticator requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

Ports can have the following authorization states:

Force authorized
Disables 802.1X port-based authentication and transitions to the authorized state without requiring any authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This authorization state is the default.

Force unauthorized
Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The authenticator cannot provide authentication services to the client through the interface.

Auto
Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received from the supplicant. The authenticator requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each supplicant that attempts to access the network is uniquely identified by the authenticator by using the supplicant's MAC address.

If the supplicant is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated supplicant are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the authenticator can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and the supplicant is not granted network access.

When a supplicant logs off, it sends an EAPOL-logoff message, which causes the authenticator port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

MAC Authentication Bypass

You can configure the Cisco NX-OS device to authorize a supplicant based on the supplicant MAC address by using the MAC authentication bypass feature. For example, you can enable this feature on interfaces configured for 802.1X that are connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the supplicant, the Cisco NX-OS device tries to authorize the client by using MAC authentication bypass.

When you enable the MAC authentication bypass feature on an interface, the Cisco NX-OS device uses the MAC address as the supplicant identity. The authentication server has a database of supplicant MAC addresses that are allowed network access. After detecting a client on the interface, the Cisco NX-OS device waits for an Ethernet packet from the client. The Cisco NX-OS device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the Cisco NX-OS device grants the client access to the network.

If an EAPOL packet is detected on the interface during the lifetime of the link, the Cisco NX-OS device determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the Cisco NX-OS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the Cisco NX-OS device does not unauthorize the client connected to the interface. When reauthentication occurs, the Cisco NX-OS device uses 802.1X authentication as the preferred reauthentication process.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*.

MAC authentication bypass interacts with the following features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.

- Port security—You can configure 802.1X authentication and port security on the same Layer 2 ports.

- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

Dynamic VLAN Assignment based on MAC-Based Authentication (MAB)

The Cisco Nexus 9000 Series switches supports dynamic VLAN assignment. After the 802.1x authentication or MAB is completed; before bringing up the port, you may want to (as part of authorization) allow the peer/host to be placed into a particular VLAN based as a result of the authentication. The RADIUS server typically indicates the desired VLAN by including tunnel attributes within the Access-Accept message. This procedure of getting the VLAN an binding it to the port constitutes to Dynamic VLAN assignment.

VLAN Assignment from RADIUS

After authentication is completed either through dot1x or MAB, the response from the RADIUS server can have dynamic VLAN information, which can be assigned to a port. This information is present in response from RADIUS server in Accept-Access message in the form of tunnel attributes. For use in VLAN assignment, the following tunnel attributes are sent:

- Tunnel-type=VLAN(13)

- Tunnel-Medium-Type=802

- Tunnel-Private-Group-ID=VLANID

All the three parameters must be received for configuring access VLAN.

Single Host and Multiple Hosts Support

The 802.1X feature can restrict traffic on a port to only one endpoint device (single-host mode) or allow traffic from multiple endpoint devices on a port (multi-host mode).

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the Cisco NX-OS device puts the port in the authorized state. When the endpoint device leaves the port, the Cisco NX-OS device put the port back into the unauthorized state. A security violation in 802.1X is defined as a detection of frames sourced from any MAC address other than the single MAC address authorized as a result of successful authentication. In this case, the interface on which this security association violation is detected (EAPOL frame from the other MAC address) will be disabled. Single host mode is applicable only for host-to-switch topology and when a single host is connected to the Layer 2 (Ethernet access port) or Layer 3 port (routed port) of the Cisco NX-OS device.

Only the first host has to be authenticated on the 802.1X port configured with multiple host mode. The port is moved to the authorized state after the successful authorization of the first host. Subsequent hosts are not required to be authorized to gain network access once the port is in the authorized state. If the port becomes unauthorized when reauthentication fails or an EAPOL logoff message is received, all attached hosts are denied access to the network. The capability of the interface to shut down upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies.

Supported Topologies

The 802.1X port-based authentication support point-to-point topology.

In this configuration, only one supplicant (client) can connect to the 802.1X-enabled authenticator (Cisco NX-OS device) port. The authenticator detects the supplicant when the port link state changes to the up state.

If a supplicant leaves or is replaced with another supplicant, the authenticator changes the port link state to down, and the port returns to the unauthorized state.

Licensing Requirements for 802.1X

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	802.1X requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

Prerequisites for 802.1X

802.1X has the following prerequisites:

- One or more RADIUS servers are accessible in the network.

8021X Guidelines and Limitations

802.1X port-based authentication has the following configuration guidelines and limitations:

- When you upgrade the Cisco Nexus Series switch to Cisco NX-OS Release 9.2(1) using the (dispruptive/non-disruptive) In-Service Software Upgrades (ISSU), you must first the disable 802.1x using the no feature dot1x command and then enable it using the feature dot1x commandfor mutli-authentication to work.

- Beginning with Cisco NX-OS Release 9.2(1), multi-authentication mode is enabled on an 802.1X port. VLAN assignment occurs successfully for the first authenticated host. Subsequent authorized (based on user credentials) data hosts are considered successfully authenticated, provided either they have no VLAN assignment or have a VLAN assignment matching the first successfully authenticated host on the port. This ensures that all successfully authenticated hosts on a port are members of the same VLAN. Flexibility of VLAN assignment is only provided to the first authenticated host.

- The Cisco Nexus 3000 Series switches support 802.1X authentication only on physical ports.

- The Cisco Nexus 3000 Series switches support 802.1X authentication on member ports of a port channel but not on the port channel itself.

- The Cisco Nexus 3000 Series switches support 802.1X authentication only on Ethernet interfaces that are in a port channel, a trunk, or an access port.

- Cisco Nexus 3000 Series switches do not support 802.1X on the following:
  - FEX ports
  - VPC ports
  - PVLAN ports
  - L3 (routed) ports

- Port security

- Ports enabled with CTS and MACsec

Note You must disable 802.1X on FEX and VPC ports, and the unsupported features.

- The Cisco Nexus 3000 Series switches do not support 802.1X authentication on port channels or subinterfaces.

- The Cisco NX-OS software does not support the following 802.1X configurations on port channel members when the members are configured for 802.1X:

  - Host mode cannot be configured in single-host mode. Only multi-host mode is supported on the member ports.

  - Single-host mode cannot be configured on member ports of a port channel. Only multi-host mode is supported on member ports of a port channel.

  - MAC authentication bypass cannot be enabled on the member ports.

  - Port security cannot be configured on the port channel.

- Member ports with and without 802.1X configuration can coexist in a port channel. However, you must ensure the identical 802.1X configuration on all the member ports in order for channeling to operate with 802.1X.

- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.

- On a 802.1X enabled port, the STP BPDUs are permitted only after a successful authentication. We recommend that you enable the 802.1X functionality only on the STP edge ports to avoid STP disputes.

- The Cisco Nexus 3000 Series switches do not support single host mode on trunk interfaces or member interfaces in a port channel.

- The Cisco Nexus 3000 series switches do not support MAC address authentication bypass on a port channel and trunk interfaces.

- The Cisco Nexus 3000 series switches do not support Dot1X on vPC ports and MCT.

- The Cisco Nexus 3000 Series switches do not support the following 802.1X hostmodes:

  - Multi authentication mode

  - Multi domain mode

- The Cisco Nexus 3000 Series switches do not support the following 802.1X protocol enhancements:

  - Critical VLAN

  - Auth failed VLAN

  - Dyanamic VLAN assignment

  - Private VLAN assignment

- Wake on LAN support

- Voice VLAN support

- Downloadable ACLs

- One-to-many logical VLAN name to ID mapping

- Web authorization

- Dynamic domain bridge assignment

- IP telephony

- Guest VLANs

- A security violation occurs when the same mac is learnt on a different VLAn with dot1x enabled on the interface.

- When you disable the dot1x feature on the Cisco Nexus 9000 Series switches, EAP packets flood in the BD.

- Configuring mac learn disable with dot1x enabled on DME enabled platform does not display the error messages.

- In Cisco Nexus Release 9.2(1), tagged EAPOL frames are processed even though the VLAN is not configured on the interface and the authentication is successful on the interface for the client.

- Secure mac learnt on orphan port is not synced on the vPC peer.

Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

Table 19: Default 802.1X Parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (**force-authorized**) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet timeout period	60 seconds (number of seconds that the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant)

Parameters	Default
Retransmission timeout period	30 seconds (number of seconds that the Cisco NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)
Maximum retransmission number	2 times (number of times that the Cisco NX-OS device will send an EAP-request/identity frame before restarting the authentication process)
Host mode	Single host
Supplicant timeout period	30 seconds (when relaying a request from the authentication server to the supplicant, the amount of time that the Cisco NX-OS device waits for a response before retransmitting the request to the supplicant)
Authentication server timeout period	30 seconds (when relaying a response from the supplicant to the authentication server, the amount of time that the Cisco NX-OS device waits for a reply before retransmitting the response to the server)

Configuring 802.1X

This section describes how to configure the 802.1X feature.

Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

Procedure

Step 1   Enable the 802.1X feature.

Step 2   Configure the connection to the remote RADIUS server.

Step 3   Enable 802.1X feature on the Ethernet interfaces.

Enabling the 802.1X Feature

You must enable the 802.1X feature on the Cisco NX-OS device before authenticating any supplicant devices.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature. The default is disabled.

Step 3	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Authentication Methods for 802.1X

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Cisco NX-OS device can perform 802.1X authentication.

Before you begin

Obtain the names or addresses for the remote RADIUS server groups.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authentication dot1x default group** *group-list* **Example:**	Specifies the RADIUS server groups to use for 802.1X authentication.

	Command or Action	Purpose
	switch(config)# aaa authentication dot1x default group rad2	The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses the global pool of RADIUS servers for authentication.
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits configuration mode.
Step 4	(Optional) **show radius-server** **Example:** switch# show radius-server	Displays the RADIUS server configuration.
Step 5	(Optional) **show radius-server group** [*group-name*] **Example:** switch# show radius-server group rad2	Displays the RADIUS server group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Controlling 802.1X Authentication on an Interface

You can control the 802.1X authentication performed on an interface. An interface can have the following 802.1X authentication states:

Auto

Enables 802.1X authentication on the interface.

Force-authorized

Disables 802.1X authentication on the interface and allows all traffic on the interface without authentication. This state is the default.

Force-unauthorized

Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot* / *port*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x port-control** {**auto** | **force-authorized** | **forced-unauthorized**}
	
	Example:
	
	`switch(config-if)# dot1x port-control`
	`auto`
		Changes the 802.1X authentication state on the interface. The default is force-authorized.

Step 4	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.

Step 6	(Optional) **show dot1x interface ethernet** *slot* / *port*
	
	Example:
	
	`switch# show dot1x interface ethernet`
	`2/1`
		Displays 802.1X feature status and configuration information for an interface.

Step 7	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring 802.1X Authentication on Member Ports

You can configure 802.1X authentication on the members of a port channel.

Note You cannot configure 802.1X authentication on the port channel itself.

There are two ways to configure 802.1X authentication on member ports: 1) by configuring 802.1X on a member port and then adding the port to a port channel or 2) by creating a port channel, adding a port to the port channel, and then configuring 802.1X on the port. The following procedure provides instructions for the first method. To configure 802.1X using the second method, use these commands:

- **interface port-channel** *channel-number*

- **interface ethernet** *slot*/*port*

- **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]

- **dot1x port-control auto**

> **Note** For more information on the above commands, see the *Cisco NX-OS Interfaces Command Reference* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	
	Example:
	
	`switch(config)# interface ethernet 7/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x port-control auto**
	
	Example:
	
	`switch(config-if)# dot1x port-control`
	`auto`
		Changes the 802.1X authentication state on the interface.

Step 4	[**no**] **switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 port or, if you use the **no** keyword, as a Layer 3 port.

Step 5	**dot1x host-mode multi-host**
	
	Example:
	
	`switch(config-if)# dot1x host-mode`
	`multi-host`
		Enables multiple hosts mode for the interface. This command is required in order to add a port to a port channel.

	Command or Action	Purpose
Step 6	**channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}] **Example:** `switch(config-if)# channel-group 5 force`	Configures the port in a channel group and sets the mode. The channel number range is from 1 to 4096.The Cisco NX-OS software creates the port channel associated with this channel group if the port channel does not already exist. The optional **force** keyword allows you to force an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group. **Note** To remove an 802.1X-enabled port from a port channel, use the **no channel-group** *channel-number* command.

Step 7	**exit**
	
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 8	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 9	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 10	(Optional) **show dot1x interface ethernet** *slot*/*port* **Example:** `switch# show dot1x interface ethernet 7/1`	Displays 802.1X feature status and configuration information for an interface.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating or Removing an Authenticator PAE on an Interface

You can create or remove the 802.1X authenticator port access entity (PAE) instance on an interface.

Note By default, the Cisco NX-OS software creates the authenticator PAE instance on the interface when you enable 802.1X on an interface.

Before you begin

Enable the 802.1X feature.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show dot1x interface ethernet** *slot/port* **Example:** `switch# show do1x interface ethernet 2/1`	Displays the 802.1X configuration on the interface.

Step 3	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 4	[**no**] **dot1x pae authenticator** **Example:** `switch(config-if)# dot1x pae authenticator`	Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface. **Note** If an authenticator PAE already exists on the interface the **dot1x pae authentication** command does not change the configuration on the interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Periodic Reauthentication for an Interface

You can enable periodic 802.1X reauthentication on an interface and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication defaults to the global value.

✎

| **Note** | During the reauthentication process, the status of an already authenticated supplicant is not disrupted. |

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x re-authentication**
	
	Example:
	
	`switch(config-if)# dot1x`
	`re-authentication`
		Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.

Step 4	(Optional) **dot1x timeout re-authperiod** *seconds*
	
	Example:
	
	`switch(config-if)# dot1x timeout`
	`re-authperiod 3300`
		Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535.
		
		Note This command affects the behavior of the Cisco NX-OS device only if you enable periodic reauthentication on the interface.

Step 5	**exit**
	
	Example:
	
	`switch(config-if)# exit`
	`switch(config)#`
		Exits configuration mode.

Step 6	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.

Step 7	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Manually Reauthenticating Supplicants

You can manually reauthenticate the supplicants for the entire Cisco NX-OS device or for an interface.

> **Note**   During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	**dot1x re-authenticate** [**interface** *slot*/*port*] **Example:** ``` switch# dot1x re-authenticate interface 2/1 ```	Reauthenticates the supplicants on the Cisco NX-OS device or on an interface.

Changing 802.1X Authentication Timers for an Interface

You can change the following 802.1X authentication timers on the Cisco NX-OS device interfaces:

Quiet-period timer
When the Cisco NX-OS device cannot authenticate the supplicant, the switch remains idle for a set period of time and then tries again. The quiet-period timer value determines the idle period. An authentication failure might occur because the supplicant provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default. The default is the value of the global quiet period timer. The range is from 1 to 65535 seconds.

Rate-limit timer
The rate-limit period throttles EAPOL-Start packets from supplicants that are sending too many EAPOL-Start packets. The authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated for the rate-limit period duration. The default value is 0 seconds and the authenticator processes all EAPOL-Start packets. The range is from 1 to 65535 seconds.

Switch-to-authentication-server retransmission timer for Layer 4 packets
The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Cisco NX-OS device waits a set period of time and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP response frames
The supplicant responds to the EAP-request/identity frame from the Cisco NX-OS device with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP request frames
The supplicant notifies the Cisco NX-OS device it that received the EAP request frame. If the authenticator does not receive this notification, it waits a set period of time and then retransmits the frame. The default is the value of the global retransmission period timer. The range is from 1 to 65535 seconds.

Inactive period timeout

When the Cisco NX-OS device remains inactive for a set period of time. The timeout inactivity-period value determines the inactive period. The recommended minimum value is 1800 seconds. You must ensure that the value is less than the value of the re-authentication time.

Note You should change the default values only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain supplicants and authentication servers.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	(Optional) **dot1x timeout quiet-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`quiet-period 25`
		Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

Step 4	(Optional) **dot1x timeout ratelimit-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`ratelimit-period 10`
		Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.

Step 5	(Optional) **dot1x timeout server-timeout** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`server-timeout 60`
		Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.

Step 6	(Optional) **dot1x timeout supp-timeout** *seconds*	Sets the number of seconds that the Cisco NX-OS device waits for the supplicant to respond to an EAP request frame before the

	Command or Action	Purpose
	Example: `switch(config-if)# dot1x timeout supp-timeout 20`	Cisco NX-OS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 7	(Optional) **dot1x timeout tx-period** *seconds* **Example:** `switch(config-if)# dot1x timeout tx-period 40`	Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 8	(Optional) **dot1x timeout inactivity-period** *seconds* **Example:** `switch(config-if)# dot1x timeout inactivity-period 1800`	Sets the number of seconds the switch can remain inactive. The recommended minimum value is1800 seconds.

Step 9	**exit**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 10	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays the 802.1X configuration.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling MAC Authentication Bypass

You can enable MAC authentication bypass on an interface that has no supplicant connected.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x mac-auth-bypass** [**eap**] **Example:** `switch(config-if)# dot1x mac-auth-bypass`	Enables MAC authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the Cisco NX-OS device to use EAP for authorization.

Step 4	**exit**
	
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Single Host or Multiple Hosts Mode

You can enable single host or multiple hosts mode on an interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# `**`configure terminal`** `switch(config)#`	Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

		Command or Action	Purpose
Step 3		**dot1x host-mode** {**multi-host**	**single-host**}	Configures the host mode. The default is single-host.
		Example:	
		switch(config-if)# **dot1x host-mode multi-host**	**Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.
Step 4		**dot1x host-mode multi-auth**	Configures the multiple authentication mode. The port is authorized only on a successful authentication of either EAP or MAB or a combination of both. Failure to authenticate will restrict network access.
		Example:	
		switch(config-if)# **dot1x host-mode multi-auth**	authentication either EAP or MAB
Step 5		**exit**	Exits configuration mode.
		Example:	
		switch(config-if)# **exit** switch(config)#	
Step 6		(Optional) **show dot1x all**	Displays all 802.1X feature status and configuration information.
		Example:	
		switch# **show dot1x all**	
Step 7		(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.
		Example:	
		switch(config)# **copy running-config startup-config**	

Disabling 802.1X Authentication on the Cisco NX-OS Device

You can disable 802.1X authentication on the Cisco NX-OS device. By default, the Cisco NX-OS software enables 802.1X authentication after you enable the 802.1X feature. However, when you disable the 802.1X feature, the configuration is removed from the Cisco NX-OS device. The Cisco NX-OS software allows you to disable 802.1X authentication without losing the 802.1X configuration.

Note   When you disable 802.1X authentication, the port mode for all interfaces defaults to force-authorized regardless of the configured port mode. When you reenable 802.1X authentication, the Cisco NX-OS software restores the configured port mode on the interfaces.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no dot1x system-auth-control**
	
	Example:
	`switch(config)# no dot1x`
	`system-auth-control`
		Disables 802.1X authentication on the Cisco NX-OS device. The default is enabled.
		
		Note Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Cisco NX-OS device.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Disabling the 802.1X Feature

You can disable the 802.1X feature on the Cisco NX-OS device.

When you disable 802.1X, all related configurations are automatically discarded. The Cisco NX-OS software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Cisco NX-OS System Management Configuration Guide* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	**Purpose**
	switch# **configure terminal** switch(config)#	
Step 2	**no feature dot1x** **Example:** switch(config)# **no feature dot1x**	Disables 802.1X. **Caution** Disabling the 802.1X feature removes all 802.1X configuration.
Step 3	**exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch# **copy running-config** **startup-config**	Copies the running configuration to the startup configuration.

Resetting the 802.1X Interface Configuration to the Default Values

You can reset the 802.1X configuration for an interface to the default values.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	 Example:
	 switch(config)# interface ethernet 2/1
	 switch(config-if)
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x default** **Example:** switch(config-if)# dot1x default	Reverts to the 802.1X configuration default values for the interface.
Step 4	**exit** **Example:** switch(config-if)# exit switch(config)#	Exits configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Setting the Maximum Authenticator-to-Supplicant Frame for an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits authentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x max-req** *count* **Example:** `switch(config-if)# dot1x max-req 3`	Changes the maximum authorization request retry count. The default is 2 times and the range is from 1 to 10. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 4	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits interface configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling RADIUS Accounting for 802.1X Authentication

You can enable RADIUS accounting for the 802.1X authentication activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**dot1x radius-accounting** **Example:** `switch(config)# dot1x radius-accounting`	Enables RADIUS accounting for 802.1X. The default is disabled.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Accounting Methods for 802.1X

You can enable AAA accounting methods for the 802.1X feature.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting dot1x default group** *group-list*	Configures AAA accounting for 802.1X. The default is disabled.
		The *group-list* argument consists of a space-delimited list of group names. The group names are the following:
		• **radius**—For all configured RADIUS servers.
		• *named-group*—Any configured RADIUS server group name.
Step 3	**exit**	Exits configuration mode.
Step 4	(Optional) **show aaa accounting**	Displays the AAA accounting configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the 802.1x feature:

```
switch# configure terminal
switch(config)# aaa accounting dot1x default group radius
switch(config)# exit
switch# show aaa accounting
switch# copy running-config startup-config
```

Setting the Maximum Reauthentication Retry Count on an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits reauthentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x max-reauth-req** *retry-count*
	
	Example:
	`switch(config-if)# dot1x max-reauth-req`
	` 3`
		Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10.

Step 4	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits interface configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Verifying the 802.1X Configuration

To display 802.1X information, perform one of the following tasks:

Command	Purpose
show dot1x	Displays the status of the 802.1X.
show dot1x all [**details** | **statistics** | **summary**]	Displays the status and all related information of the 802.1X feature.
show dot1x interface ethernet *slot*/*port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an Ethernet interface.
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.

Command	Purpose
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 3000 Series NX-OS Security Command Reference*.

802.1X Support for VXLAN EVPN

This section describes how to configure configures 802.1X for VXLAN EVPN.

Guidelines and Limitations for 802.1X Support for VXLAN EVPN

The following are the guidelines and limitations for 802.1X support for VXLAN EVPN:

•

- Port channel interfaces or the member ports of the port channel are not supported.

- vPC ports are not supported.

- The current support of the feature uses regular and dynamic EVPN updates on the BGP-EVPN control plane for 802.1X secure MAC updates. As a result, we cannot prevent the move across EVPN even if the global policy is "dot1x mac-move deny".

- Ensure that the "dot1x mac-move" policy is configured the same across the fabric. There is no configuration validation across the nodes, hence it could lead to unexpected behavior if the configuration policy is not in sync.

- The local to remote MAC moves behavior for the deny and permit modes is permitted. Therefore, the MAC move is permitted even if the deny mode is enabled.

- Ensure that the dot1x and the port-security ports use different VLANs. The same VLAN cannot be assigned to both ports.

- Dot1x is not VLAN aware and hence having the same MAC in two different VLANs is not possible. Depending on the mac-move mode that is selected, either the MAC is moved to a new VLAN or it is denied.

- You cannot configure static and secure MAC together.

- The Cisco Nexus 3164Q-40GE switch is not supported.

- RADIUS change of Authorization is supported for VXLAN EVPN.

- The recommended re-authentication time interval for a scale setup is the default value, which is 3600 seconds.

Configuring 802.1X Support for VXLAN EVPN

This procedure configures 802.1X for VXLAN EVPN.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# **configure terminal**	Enters global configuration mode.
Step 2	**feature dot1x** **Example:** switch(config)# **feature dot1x**	Enables the 802.1X feature. The default is disabled.
Step 3	**dot1x mac-move {permit | deny}** **Example:** switch(config)# **dot1x mac-move permit**	The **deny** parameters denies MAC moves. The **permit** parameter permits MAC moves.

Step 4	(Optional) **show running-config dot1x all**
	
	Example:
	
	switch(config)# **show running-config dot1x all**
	
	!Command: show running-config dot1x all
	!No configuration change since last restart
	!Time: Thu Sep 20 10:22:58 2018
	
	version 9.2(2) Bios:version 07.64
	feature dot1x
	
	dot1x system-auth-control
	dot1x mac-move deny
	
	interface Ethernet1/1
	  dot1x host-mode multi-auth
	  dot1x pae authenticator
	  dot1x port-control auto
	  no dot1x re-authentication
	  dot1x max-req 1
	  dot1x max-reauth-req 2
	  dot1x timeout quiet-period 60
	  dot1x timeout re-authperiod 3600
	  dot1x timeout tx-period 1
	  dot1x timeout server-timeout 30
	  dot1x timeout ratelimit-period 0
	  dot1x timeout supp-timeout 30
	  dot1x timeout inactivity-period 0
	  dot1x mac-auth-bypass
	
	interface Ethernet1/33
	  dot1x host-mode multi-auth
	  dot1x pae authenticator
	  dot1x port-control auto
	  no dot1x re-authentication
	  dot1x max-req 1
	  dot1x max-reauth-req 2
	  dot1x timeout quiet-period 60
	  dot1x timeout re-authperiod 3600
		Displays the 802.1X configuration.

Command or Action	Purpose

`dot1x timeout tx-period 1`
`dot1x timeout server-timeout 30`
`dot1x timeout ratelimit-period 0`
`dot1x timeout supp-timeout 30`
`dot1x timeout inactivity-period 0`
`dot1x mac-auth-bypass`	

Verifying the 802.1X Support for VXLAN EVPN

To display the 802.1X Support for VXLAN ENPN configuration information, enter one of the following commands:

Command	Purpose
show running-config dot1x all	Displays dot1x running configuration.
show dot1x all summary	Displays the interface status.
show dot1x	Displays the default settings.
show dot1x all	Displays additional interface detail.

Example of show running-config dot1x all command

```
switch# show running-config dot1x all
!Command: show running-config dot1x all
!No configuration change since last restart
!Time: Thu Sep 20 10:22:58 2018

version 9.2(2) Bios:version 07.64
feature dot1x

dot1x system-auth-control
dot1x mac-move deny


interface Ethernet1/1
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
  dot1x max-req 1
  dot1x max-reauth-req 2
  dot1x timeout quiet-period 60
  dot1x timeout re-authperiod 3600
  dot1x timeout tx-period 1
  dot1x timeout server-timeout 30
  dot1x timeout ratelimit-period 0
  dot1x timeout supp-timeout 30
  dot1x timeout inactivity-period 0
  dot1x mac-auth-bypass

interface Ethernet1/33
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
```

```
dot1x max-req 1
dot1x max-reauth-req 2
dot1x timeout quiet-period 60
dot1x timeout re-authperiod 3600
dot1x timeout tx-period 1
dot1x timeout server-timeout 30
dot1x timeout ratelimit-period 0
dot1x timeout supp-timeout 30
dot1x timeout inactivity-period 0
dot1x mac-auth-bypass
```

Example of the show dot1x all summary command

```
switch# show dot1x all summary

           Interface     PAE            Client        Status
------------------------------------------------------------------
        Ethernet1/1     AUTH             none      UNAUTHORIZED

           Interface     PAE            Client        Status
------------------------------------------------------------------
       Ethernet1/33    AUTH   00:16:5A:4C:00:07    AUTHORIZED
                              00:16:5A:4C:00:06    AUTHORIZED
                              00:16:5A:4C:00:05    AUTHORIZED
                              00:16:5A:4C:00:04    AUTHORIZED
switch#
switch# show mac address-table vlan 10
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
    VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10     0016.5a4c.0004   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0005   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0006   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0007   secure   -         T      F    Eth1/33

switch#
switch# show mac address-table vlan 10 (VPC-PEER)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
    VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10     0016.5a4c.0004   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0005   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0006   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0007   secure   -         T      F    vPC Peer-Link

switch#
switch# show mac address-table vlan 10 (RVTEP)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
    VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C   10     0016.5a4c.0004   dynamic  0         F      F    nve1(67.67.67.67)
C   10     0016.5a4c.0005   dynamic  0         F      F    nve1(67.67.67.67)
C   10     0016.5a4c.0006   dynamic  0         F      F    nve1(67.67.67.67)
C   10     0016.5a4c.0007   dynamic  0         F      F    nve1(67.67.67.67)
```

Example of the show dot1x command

```
switch# show dot1x
          Sysauthcontrol Enabled
   Dot1x Protocol Version 2
                Mac-Move Deny
```

Example of the show dot1x all command

```
switch# show dot1x  all
          Sysauthcontrol Enabled
   Dot1x Protocol Version 2
                Mac-Move Deny

Dot1x Info for Ethernet1/1
-----------------------------------
                       PAE = AUTHENTICATOR
              PortControl = AUTO
                  HostMode = MULTI AUTH
         ReAuthentication = Disabled
               QuietPeriod = 60
            ServerTimeout = 30
              SuppTimeout = 30
             ReAuthPeriod = 3600 (Locally configured)
                ReAuthMax = 2
                   MaxReq = 1
                 TxPeriod = 1
          RateLimitPeriod = 0
         InactivityPeriod = 0
          Mac-Auth-Bypass = Enabled

Dot1x Info for Ethernet1/33
-----------------------------------
                       PAE = AUTHENTICATOR
              PortControl = AUTO
                  HostMode = MULTI AUTH
         ReAuthentication = Disabled
               QuietPeriod = 60
            ServerTimeout = 30
              SuppTimeout = 30
             ReAuthPeriod = 3600 (Locally configured)
                ReAuthMax = 2
                   MaxReq = 1
                 TxPeriod = 1
          RateLimitPeriod = 0
         InactivityPeriod = 0
          Mac-Auth-Bypass = Enabled
```

Monitoring 802.1X

You can display the statistics that the Cisco NX-OS device maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	**show dot1x** {**all** | **interface ethernet** *slot*/*port*} **statistics** **Example:** switch# **show dot1x all statistics**	Displays the 802.1X statistics.

Configuration Example for 802.1X

The following example shows how to configure 802.1X for an access port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
```

The following example shows how to configure 802.1X for a trunk port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
dot1x host-mode multi-host
```

> **Note** Repeat the **dot1x pae authenticator** and **dot1x port-control auto** commands for all interfaces that require 802.1X authentication.

Additional References for 802.1X

This section includes additional information related to implementing 802.1X.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	Cisco NX-OS Licensing Guide
Command reference	Cisco Nexus 9000 Series NX-OS Security Command Reference
VRF configuration	Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide

Standards

Standards	Title
IEEE Std 802.1X- 2004 (Revision of IEEE Std 802.1X-2001)	*802.1X IEEE Standard for Local and Metropolitan Area Networks Port-Based Network Access Control*
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 3580	*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*

Configuring Unicast RPF

This chapter describes how to configure rate limits for egress traffic on Cisco NX-OS devices and includes the following sections:

About Unicast RPF

The Unicast RPF feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IPv4 or IPv6 source addresses into a network by discarding IPv4 or IPv6 packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IPv4 or IPv6 addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

When you enable Unicast RPF on an interface, the switch examines all ingress packets received on that interface to ensure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This examination of source addresses relies on the Forwarding Information Base (FIB).

Note Unicast RPF is an ingress function and is applied only on the ingress interface of a switch at the upstream end of a connection.

Unicast RPF verifies that any packet received at a switch interface arrives on the best return path (return route) to the source of the packet by doing a reverse lookup in the FIB. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same

interface from which the packet was received, the source address might have been modified by the attacker. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

✎

Note With Unicast RPF, all equal-cost "best" return paths are considered valid, which means that Unicast RPF works where multiple return paths exist, if each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are being used and unequal candidate paths back to the source IP address exist.

Unicast RPF Process

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB that matches the route to the receiving interface. Static routes, network statements, and dynamic routing add routes to the FIB.

- IP source addresses at the receiving interface must match the routing entry for the interface.

- Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

You can use Unicast RPF for downstream networks, even if the downstream network has other connections to the Internet.

⚠

Caution Be careful when using optional BGP attributes, such as weight and local preference, because an attacker can modify the best path back to the source address. Modification would affect the operation of Unicast RPF.

When a packet is received at the interface where you have configured Unicast RPF and ACLs, the Cisco NX-OS software performs the following actions:

Procedure

Step 1 Checks the input ACLs on the inbound interface.

Step 2 Uses Unicast RPF to verify that the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 Conducts a FIB lookup for packet forwarding.

Step 4 Checks the output ACLs on the outbound interface.

Step 5 Forwards the packet.

Licensing Requirements for Unicast RPF

Product	License Requirement
Cisco NX-OS	Unicast RPF requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for Unicast RPF

Unicast RPF has the following configuration guidelines and limitations:

- The following platforms support uRPF:
    - Cisco Nexus 3100 platform switches in N3K mode.
    - Cisco Nexus 3132Q-V, 31108PC-V, and 31108TC-V switches.
    - Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches and Cisco Nexus 3100 platform switches in N9K mode.

- For Cisco Nexus 3164Q switches, Unicast RPF is supported only with the non-hierarchical routing mode.

- You must apply Unicast RPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.

- The further downstream that you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.

- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources means that the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances are of tracing the source of an attack.

- Unicast RPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure Unicast RPF at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

- You can use Unicast RPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.

- Do not use Unicast RPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure Unicast RPF only where there is natural or configured symmetry. Do not configure strict Unicast RPF.

- Unicast RPF allows packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.

- If the route to the source IP is Equal Cost Multi Path route with more than 8 members, the Unicast RPF check performed on the ingress interface cannot be of the strict mode and by default, will be of loose mode.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for Unicast RPF

This table lists the default settings for Unicast RPF parameters.

Table 20: Default Unicast RPF Parameter Settings

Parameters	Default
Unicast RPF	Disabled for all switches except the Cisco Nexus 3100 platform switches in N3K mode

Configuring Unicast RPF

You can configure one the following Unicast RPF modes on an ingress interface:

Strict Unicast RPF mode
A strict mode check is successful when Unicast RPF finds a match in the FIB for the packet source address and the ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match. If this check fails, the packet is discarded. You can use this type of Unicast RPF check where packet flows are expected to be symmetrical.

Loose Unicast RPF mode
A loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet is received is not required to match any of the interfaces in the FIB result.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **system urpf disable**	Enables Unicast RPF on the switch.

	Command or Action	Purpose
	Example: switch(config)# system urpf disable	**Note** You must reload the Cisco NX-OS box to apply the Unicast RPF configuration.
Step 3	**interface ethernet** *slot/port* **Example:** switch(config)# interface ethernet 2/3 switch(config-if)#	Specifies an Ethernet interface and enters interface configuration mode.
Step 4	{**ip**	**ipv6**} **address** *ip-address/length* **Example:** switch(config-if)# ip address 172.23.231.240/23	Specifies an IPv4 or IPv6 address for the interface.
Step 5	{**ip**	**ipv6**} **verify unicast source reachable-via** {**any** [**allow-default**]	**rx**} **Example:**	Configures Unicast RPF on the interface for both IPv4 and IPv6.

Command or Action	Purpose	
`switch(config-if)# ip verify unicast source reachable-via any`	**Note**	• When you enable Unicast RPF for IPv4 or IPv6 (using the **ip** or **ipv6** keyword), Unicast RPF is enabled for both IPv4 and IPv6.
	Note	You can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. When you configure one version, all the mode changes must be done by this version and all other versions will be blocked by that interface.
		• The **any** keyword specifies loose Unicast RPF.
		• If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification.
	Note	• The **allow-default** keyword is not applicable in the ALPM routing mode.
		• The source address lookup (in case of a loose Unicast RPF check) does not match the default route if you do not specify the **allow-default** keyword.
		• The **rx** keyword specifies strict Unicast RPF.

	Command or Action	Purpose

Step 6	**exit**
	
	Example:
	
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 7	(Optional) **show ip interface ethernet** *slot/port* **Example:** `switch(config)# show ip interface` `ethernet 1/54	grep -i "unicast reverse` ` path forwarding"` `IP unicast reverse path forwarding: none`	Displays the IP information for an interface and verifies if Unicast RPF is enabled.

Step 8	(Optional) **show running-config interface ethernet** *slot/port*
	
	Example:
	
	`switch(config)# show running-config`
	`interface ethernet 2/3`
		Displays the configuration for an interface in the running configuration.

Step 9	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuration Examples for Unicast RPF

The following example shows how to configure loose Unicast RPF for IPv4 packets:

```
interface Ethernet2/3
ip address 172.23.231.240/23
ip verify unicast source reachable-via any allow-default
```

The following example shows how to configure loose Unicast RPF for IPv6 packets:

```
interface Ethernet2/3
ipv6 address 2001:0DB8:c18:1::3/64
ipv6 verify unicast source reachable-via any allow-default
```

The following example shows how to configure strict Unicast RPF for IPv4 packets:

```
interface Ethernet2/2
ip address 172.23.231.240/23
ip verify unicast source reachable-via rx
```

The following example shows how to configure strict Unicast RPF for IPv6 packets:

```
interface Ethernet2/2
ipv6 address 2001:0DB8:c18:1::3/64
ipv6 verify unicast source reachable-via rx
```

Verifying the Unicast RPF Configuration

To display Unicast RPF configuration information, perform one of the following tasks:

Command	Purpose
show running-config interface ethernet *slot*/*port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.
show running-config ipv6 [**all**]	Displays the IPv6 configuration in the running configuration.
show startup-config interface ethernet *slot*/*port*	Displays the interface configuration in the startup configuration.
show ip interface ethernet *slot*/*port*	Displays the IP information for an interface and verifies if the unicast RPF is enabled or disabled.
show startup-config ip	Displays the IP configuration in the startup configuration.

Additional References for Unicast RPF

This section includes additional information related to implementing unicast RPF.

Related Documents

Related Topic	Document Title
Data Management Engine (DME)-ized commands	Cisco Nexus 3000 and 9000 Series NX-API REST SDK User Guide and API Reference

Configuring Control Plane Policing

This chapter contains the following sections:

Information About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic destined to any of the IP addresses of the router or Layer 3 switch. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, which can be perpetrated either inadvertently or maliciously, typically involve high rates of traffic destined to the supervisor module or CPU itself.

The supervisor module divides the traffic that it manages into three functional components or planes:

Data plane
> Handles all the data traffic. The basic functionality of a Cisco NX-OS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.

Control plane
> Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

Management plane
> Runs the components meant for Cisco NX-OS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. For example, excessive traffic to the supervisor module could overload and slow down the performance of the entire Cisco NX-OS device. Another example is a DoS attack on the supervisor module that could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks are as follows:

- Internet Control Message Protocol (ICMP) echo requests

- IP fragments

- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)

- High route processor or switch processor CPU utilization

- Route flaps due to loss of routing protocol updates or keepalives

- Unstable Layer 2 topology

- Slow or unresponsive interactive sessions with the CLI

- Processor resource exhaustion, such as the memory and buffers

- Indiscriminate drops of incoming packets

⚠️

Caution   It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

Control Plane Protection

To protect the control plane, the Cisco NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Cisco NX-OS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Control Plane Packet Types

Different types of packets can reach the control plane:

Receive packets

Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

Exception packets

Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

Redirected packets

Packets that are redirected to the supervisor module. Features such as Dynamic Host Configuration Protocol (DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.

Glean packets

If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco NX-OS device. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the supervisor module receives these packets.

Classification for CoPP

For effective protection, the Cisco NX-OS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set. You configure packet classifications and rate controlling policies using class-maps and policy-maps.

The following parameters can be used to classify a packet:

- Source IP address

- Destination IP address

- Source port

- Destination port

- Layer 4 protocol

Rate Controlling Mechanisms

Once the packets are classified, the Cisco NX-OS device has different mechanisms to control the rate at which packets arrive at the supervisor module.

The policing rate is specified in terms of packets per second (PPS). Each classified flow can be policed individually by specifying a policing rate limit in PPS.

CoPP Policy Templates

When you bring up your Cisco NX-OS device for the first time, the Cisco NX-OS software installs the default copp-system-policy to protect the supervisor module from DoS attacks. You can choose the CoPP policy template for your deployment scenario by specifying CoPP policy options from the initial setup utility:

- Default—Layer 2 and Layer 3 policy which provides a good balance of policing between switched and routed traffic bound to CPU.

- Layer 2—Layer 2 policy which gives more preference to the Layer 2 traffic (eg BPDU) bound to the CPU

- Layer 3—Layer 3 policy which gives more preference to the Layer 3 traffic (eg BGP, RIP, OSPF etc) bound to the CPU

If you do not select an option or choose not to execute the setup utility, the Cisco NX-OS software applies the Default policing. Cisco recommends starting with the default policy and later modifying the CoPP policies as required.

The default copp-system-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements.

You can switch across default, Layer 2 and Layer 3 templates by entering the setup utility again using the setup command.

Default CoPP Policy

This policy is applied to the switch by default. It has the classes with police rates that should suit most network installations. You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the default CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
```

```
class copp-s-ip-options
  police pps 100
class copp-s-ip-nat
  police pps 100
class copp-s-ipmcmiss
  police pps 400
class copp-s-ipmc-g-hit
  police pps 400
class copp-s-ipmc-rpf-fail-g
  police pps 400
class copp-s-ipmc-rpf-fail-sg
  police pps 400
class copp-s-dhcpreq
  police pps 300
class copp-s-dhcpresp
  police pps 300
class copp-s-igmp
  police pps 400
class copp-s-routingProto2
  police pps 1300
class copp-s-eigrp
  police pps 200
class copp-s-pimreg
  police pps 200
class copp-s-pimautorp
  police pps 200
class copp-s-routingProto1
  police pps 1000
class copp-s-arp
  police pps 200
class copp-s-ptp
  police pps 1000
class copp-s-bpdu
  police pps 12000
class copp-s-cdp
  police pps 400
class copp-s-lacp
  police pps 400
class copp-s-lldp
  police pps 200
class copp-icmp
  police pps 200
class copp-telnet
  police pps 500
class copp-ssh
  police pps 500
class copp-snmp
  police pps 500
class copp-ntp
  police pps 100
class copp-tacacsradius
  police pps 400
class copp-stftp
  police pps 400
class copp-ftp
  police pps 100
class copp-http
  police pps 100
```

This is the default CoPP policy profile for Cisco Nexus 34180YC.

```
sh policy-map int control-plane
      Control Plane

        Service-policy  input: copp-system-p-policy-strict
```

```
class-map copp-system-p-class-l3uc-data (match-any)
  match exception glean
  set cos 1
  police cir 250 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-critical (match-any)
  match access-group name copp-system-p-acl-bgp
  match access-group name copp-system-p-acl-rip
  match access-group name copp-system-p-acl-vpc
  match access-group name copp-system-p-acl-bgp6
  match access-group name copp-system-p-acl-ospf
  match access-group name copp-system-p-acl-rip6
  match access-group name copp-system-p-acl-eigrp
  match access-group name copp-system-p-acl-ospf6
  match access-group name copp-system-p-acl-eigrp6
  match access-group name copp-system-p-acl-auto-rp
  match access-group name copp-system-p-acl-mac-l3-isis
  set cos 7
  police cir 19000 pps , bc 128 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-important (match-any)
  match access-group name copp-system-p-acl-hsrp
  match access-group name copp-system-p-acl-vrrp
  match access-group name copp-system-p-acl-hsrp6
  match access-group name copp-system-p-acl-vrrp6
  match access-group name copp-system-p-acl-mac-lldp
  set cos 6
  police cir 3000 pps , bc 256 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-openflow (match-any)
  match access-group name copp-system-p-acl-openflow
  set cos 5
  police cir 2000 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-multicast-router (match-any)
  match access-group name copp-system-p-acl-pim
  match access-group name copp-system-p-acl-msdp
  match access-group name copp-system-p-acl-pim6
  match access-group name copp-system-p-acl-pim-reg
  match access-group name copp-system-p-acl-pim6-reg
  match access-group name copp-system-p-acl-pim-mdt-join
  set cos 6
  police cir 3000 pps , bc 128 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-multicast-host (match-any)
  match access-group name copp-system-p-acl-mld
  set cos 1
  police cir 2000 pps , bc 128 packets
```

```
          module 1 :
            transmitted 0 packets;
            dropped 0 packets;

class-map copp-system-p-class-l3mc-data (match-any)
  match exception multicast rpf-failure
  match exception multicast dest-miss
  set cos 1
  police cir 3000 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-normal (match-any)
  match access-group name copp-system-p-acl-mac-dot1x
  match protocol arp
  set cos 1
  police cir 1500 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-ndp (match-any)
  match access-group name copp-system-p-acl-ndp
  set cos 6
  police cir 1500 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-normal-dhcp (match-any)
  match access-group name copp-system-p-acl-dhcp
  match access-group name copp-system-p-acl-dhcp6
  set cos 1
  police cir 300 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-normal-dhcp-relay-response (match-any)
  match access-group name copp-system-p-acl-dhcp-relay-response
  match access-group name copp-system-p-acl-dhcp6-relay-response
  set cos 1
  police cir 400 pps , bc 64 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-normal-igmp (match-any)
  match access-group name copp-system-p-acl-igmp
  set cos 3
  police cir 6000 pps , bc 64 packets
  module 1 :
    transmitted 0 packets;
    dropped 0 packets;

class-map copp-system-p-class-redirect (match-any)
  match access-group name copp-system-p-acl-ptp
  match access-group name copp-system-p-acl-ptp-l2
  match access-group name copp-system-p-acl-ptp-uc
  set cos 1
  police cir 1500 pps , bc 32 packets
  module 1 :
    transmitted 0 packets;
```

```
                                  dropped 0 packets;

            class-map copp-system-p-class-exception (match-any)
              match exception ip option
              match exception ip icmp unreachable
              match exception ipv6 option
              match exception ipv6 icmp unreachable
              set cos 1
              police cir 50 pps , bc 32 packets
              module 1 :
                transmitted 0 packets;
                dropped 0 packets;

            class-map copp-system-p-class-exception-diag (match-any)
              match exception ttl-failure
              match exception mtu-failure
              set cos 1
              police cir 50 pps , bc 32 packets
              module 1 :
                transmitted 0 packets;
                dropped 0 packets;

            class-map copp-system-p-class-management (match-any)
              match access-group name copp-system-p-acl-ftp
              match access-group name copp-system-p-acl-ntp
              match access-group name copp-system-p-acl-ssh
              match access-group name copp-system-p-acl-http
              match access-group name copp-system-p-acl-ntp6
              match access-group name copp-system-p-acl-sftp
             match access-group name copp-system-p-acl-snmp
              match access-group name copp-system-p-acl-ssh6
              match access-group name copp-system-p-acl-tftp
              match access-group name copp-system-p-acl-https
              match access-group name copp-system-p-acl-snmp6
              match access-group name copp-system-p-acl-tftp6
              match access-group name copp-system-p-acl-radius
              match access-group name copp-system-p-acl-tacacs
              match access-group name copp-system-p-acl-telnet
              match access-group name copp-system-p-acl-radius6
              match access-group name copp-system-p-acl-tacacs6
              match access-group name copp-system-p-acl-telnet6
              set cos 2
              police cir 3000 pps , bc 512000 packets
              module 1 :
                transmitted 0 packets;
                dropped 0 packets;

            class-map copp-system-p-class-monitoring (match-any)
              match access-group name copp-system-p-acl-icmp
              match access-group name copp-system-p-acl-icmp6
              match access-group name copp-system-p-acl-traceroute
              set cos 1
              police cir 300 pps , bc 128 packets
              module 1 :
                transmitted 0 packets;
                dropped 0 packets;

            class-map copp-system-p-class-l2-unpoliced (match-any)
              match access-group name copp-system-p-acl-mac-stp
              match access-group name copp-system-p-acl-mac-lacp
              match access-group name copp-system-p-acl-mac-cfsoe
              match access-group name copp-system-p-acl-mac-sdp-srp
              match access-group name copp-system-p-acl-mac-l2-tunnel
              match access-group name copp-system-p-acl-mac-cdp-udld-vtp
```

```
      set cos 7
      police cir 20000 pps , bc 8192 packets
      module 1 :
        transmitted 0 packets;
        dropped 0 packets;

  class-map copp-system-p-class-undesirable (match-any)
    match access-group name copp-system-p-acl-undesirable
    match exception multicast sg-rpf-failure
    set cos 0
    police cir 15 pps , bc 32 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map copp-system-p-class-fcoe (match-any)
    match access-group name copp-system-p-acl-mac-fcoe
    set cos 6
    police cir 1500 pps , bc 128 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map copp-system-p-class-nat-flow (match-any)
    match exception nat-flow
    set cos 7
    police cir 100 pps , bc 64 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map copp-system-p-class-l3mcv6-data (match-any)
    match exception multicast ipv6-rpf-failure
    match exception multicast ipv6-dest-miss
    set cos 1
    police cir 3000 pps , bc 32 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map copp-system-p-class-undesirablev6 (match-any)
    match exception multicast ipv6-sg-rpf-failure
    set cos 0
    police cir 15 pps , bc 32 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map copp-system-p-class-l2-default (match-any)
    match access-group name copp-system-p-acl-mac-undesirable
    set cos 0
    police cir 50 pps , bc 32 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;

  class-map class-default (match-any)
    set cos 0
    police cir 50 pps , bc 32 packets
    module 1 :
      transmitted 0 packets;
      dropped 0 packets;
```

Layer 2 CoPP Policy

You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the Layer 2 CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ip-options
    police pps 100
  class copp-s-ip-nat
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-ipmc-g-hit
    police pps 400
  class copp-s-ipmc-rpf-fail-g
    police pps 400
  class copp-s-ipmc-rpf-fail-sg
    police pps 400
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 1200
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
  class copp-s-pimautorp
    police pps 200
  class copp-s-routingProto1
    police pps 900
  class copp-s-arp
    police pps 200
  class copp-s-ptp
    police pps 1000
  class copp-s-bpdu
    police pps 12300
  class copp-s-cdp
    police pps 400
  class copp-s-lacp
    police pps 400
  class copp-s-lldp
    police pps 200
  class copp-icmp
    police pps 200
```

```
                        class copp-telnet
                          police pps 500
                        class copp-ssh
                          police pps 500
                        class copp-snmp
                          police pps 500
                        class copp-ntp
                          police pps 100
                        class copp-tacacsradius
                          police pps 400
                        class copp-stftp
                          police pps 400
                        class copp-ftp
                          police pps 100
                        class copp-http
                          police pps 100
```

Layer 3 CoPP Policy

You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the Layer 3 CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ip-options
    police pps 100
  class copp-s-ip-nat
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-ipmc-g-hit
    police pps 400
  class copp-s-ipmc-rpf-fail-g
    police pps 400
  class copp-s-ipmc-rpf-fail-sg
    police pps 400
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 4000
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
```

```
class copp-s-pimautorp
  police pps 200
class copp-s-routingProto1
  police pps 4000
class copp-s-arp
  police pps 200
class copp-s-ptp
  police pps 1000
class copp-s-bpdu
  police pps 6000
class copp-s-cdp
  police pps 200
class copp-s-lacp
  police pps 200
class copp-s-lldp
  police pps 200
class copp-icmp
  police pps 200
class copp-telnet
  police pps 500
class copp-ssh
  police pps 500
class copp-snmp
  police pps 500
class copp-ntp
  police pps 100
class copp-tacacsradius
  police pps 400
class copp-stftp
  police pps 400
class copp-ftp
  police pps 100
class copp-http
  police pps 100
```

Static CoPP Classes

The following are the available static CoPP classes:

- **copp-s-default**

  Catch-all CoPP class for traffic when copy-to-CPU is set for the packet and there is no match in other more specific CoPP classes for the packet.

  ```
  class-map copp-s-default (match-any)
       police pps 400
          OutPackets    0
          DropPackets   0
  ```

- **copp-s-l2switched**

  Catch-all CoPP class for Layer 2 traffic if there is no match in other explicit CoPP classes when CPU port is being selected for the packet.

  ```
  class-map copp-s-l2switched (match-any)
       police pps 200
          OutPackets    0
          DropPackets   0
  ```

- **copp-s-l3destmiss**

  Layer 3 traffic with a miss for the lookup in the hardware Layer 3 forwarding table.

```
class-map copp-s-l3destmiss (match-any)
      police pps 100
        OutPackets   0
        DropPackets  0
```

- **copp-s-glean**

Used in case of Layer 3 traffic to IP address in directly connected subnets with no ARP resolution present for the IP address to trigger ARP resolution in software.

```
class-map copp-s-glean (match-any)
      police pps 500
        OutPackets   0
        DropPackets  0
```

- **copp-s-selfip**

Default CoPP class for packets that are coming to one of the router interface's IP addresses if there is no match in other more specific CoPP classes.

```
class-map copp-s-selfIp (match-any)
      police pps 500
        OutPackets   4
        DropPackets  0
```

- **copp-s-l3mtufail**

Layer 3 packets with MTU check fail needing software processing for fragmentation or for generating ICMP message.

```
class-map copp-s-l3mtufail (match-any)
      police pps 100
        OutPackets   0
        DropPackets  0
```

- **copp-s-ttl1**

Layer 3 packets coming to one of the router's interface IP addresses and with TTL=1.

```
class-map copp-s-ttl1 (match-any)
      police pps 100
        OutPackets   0
        DropPackets  0
```

- **copp-s-ipmsmiss**

Multicast packets with lookup miss in hardware Layer 3 forwarding table for multicast forwarding lookup. These data packets can trigger the installation of the hardware forwarding table entries for hardware forwarding of multicast packets.

```
class-map copp-s-ipmcmiss (match-any)
      police pps 400
        OutPackets   0
        DropPackets  0
```

- **copp-s-l3slowpath**

Layer 3 packets that are hitting other packet exception cases that need handing in software. For example, IP option packets.

```
class-map copp-s-l3slowpath (match-any)
      police pps 100
        OutPackets   0
        DropPackets  0
```

- **copp-s-dhcpreq**

CoPP class for DHCP request packets. By default, this class is only used to program the CoPP rate for this class of packets. Copy to CPU is not enabled till DHCP snooping or relay is configured.

```
class-map copp-s-dhcpreq (match-any)
      police pps 300
        OutPackets    0
        DropPackets   0
```

- **copp-s-dai**

CoPP class for ARP inspection intercepted packets. By default, this class is only used to program the CoPP rate for this class of packets. Copy to CPU is not enabled till the IP ARP inspection feature is configured.

```
class-map copp-s-dai (match-any)
      police pps 300
        OutPackets    0
        DropPackets   0
```

- **copp-s-pimautorp**

This CoPP class is used to copy PIM auto-rp packets to the CPU (IP multicast groups 224.0.1.39 and 224.0.1.40)

```
class-map copp-s-pimautorp (match-any)
      police pps 200
        OutPackets    0
        DropPackets   0
```

- **copp-s-arp**

CoPP class for ARP and ND request and reply packets that are being copied to the CPU.

```
class-map copp-s-arp (match-any)
      police pps 200
        OutPackets    0
        DropPackets   0
```

- **copp-s-ptp**

CoPP class for Precision Time Protocol (PTP) packets.

```
class-map copp-s-ptp (match-any)
      police pps 1000
        OutPackets    0
        DropPackets   0
```

- **copp-s-vxlan**

This CoPP class is used when the NV overlay feature is configured and when packets are being copied to the CPU for remote peer IP address learning.

```
class-map copp-s-vxlan (match-any)
      police pps 1000
        OutPackets    0
        DropPackets   0
```

- **copp-s-bfd**

CoPP class for Bidirectional Forwarding Detection (BFD) packets that are being copied to the CPU (Packets with BFD protocol UDP ports, coming to router interface IP address).

```
class-map copp-s-bfd (match-any)
      police pps 600
        OutPackets    0
        DropPackets   0
```

- **copp-s-bpdu**

  CoPP class for BPDU class of packets that are being copied to the CPU. This includes STP, CDP, LLDP, LACP, and UDLD packets).

  ```
  class-map copp-s-bpdu (match-any)
       police pps 15000
          OutPackets    100738
          DropPackets   0
  ```

- **copp-s-dpss**

  CoPP class that is used for programmability features, OnePK and Openflow, when the policy is configured with punt-to-CPU action. For example, data path service set, OpenFlow punt-to-controller action.

  ```
  class-map copp-s-dpss (match-any)
       police pps 1000
          OutPackets    0
          DropPackets   0
  ```

- **copp-s-mpls**

  Used for the tap aggregation feature for MPLS label strip action. This class is used to copy the packets to the CPU to learn the MPLS label information and program for the label strip action.

  ```
  class-map copp-s-mpls (match-any)
       police pps 100
          OutPackets    0
          DropPackets   0
  ```

CoPP Class Maps

Classes within a policy are of two types:

- Static—These classes are part of every policy template and cannot be removed from the policy or CoPP configuration. Static classes would typically contain the traffic which is deemed critical to device operation and is required in the policy.

- Dynamic—These classes can be created, added or removed from a policy. Using dynamic classes, you can create classes/policing for CPU bound traffic (unicast) specific to their requirements.

Note   Classes with names copp-s-x are static classes.

ACLs can be associated with both static and dynamic classes.

Packets Per Second Credit Limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

CoPP and the Management Interface

The Cisco NX-OS device supports only hardware-based CoPP which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Licensing Requirements for CoPP

This feature does not require a license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- The PIM_IGMP class-id is set on the port only when PIM is enabled. Since there is no need to punt IGMP packets to the CPU on the Layer 3 ports when PIM is not enabled, you have to configure feature pim and enable PIM on the port to get the packets on the copp-s-igmp queue.

- Cisco recommends that you choose the default, L2, or L3 policy, depending upon your deployment scenario and later modify the CoPP policies based on observed behavior.

- First generation Nexus 3000 series switches (non -EX/FX/FX2), do not support source-based CoPP. This limitation does not exist for cloud scale ASIC-based Nexus switches

- If you observe +/- 2-5% irregularity in the traffic around 30-40s after the traffic has fully converged after fast-reload, use a higher COPP value for the ARP packets.

- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features used in your specific environment as well as the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.

- The default police packets per second (PPS) value is changed to 900 for **copp-s-bfd** command with **write erase** command and reload.

- Cisco recommends that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to use a different CoPP policy or modify the customized CoPP policy.

- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (**service-policy output copp** cannot be applied to the control plane interface).

- The creation of new CoPP policies is not supported.

- When a new CoPP class-map is inserted into the CoPP policy-map with the **insert-before** option, the order of the class-maps is retained in the running-configuration. However, after you run the **write erase** command and reload the switch, the default CoPP policy is applied, and the class-maps are rearranged in the default order. When you copy the file to the running-configuration, it becomes a modify operation

for the existing CoPP policy and the new class-maps are inserted at the end. Similarly, if there is change in the order of default class-maps in the file, it will not be effective. To preserve the order of the class-maps, copy the configuration to startup and reload.

- IPv6 and IPv4 CoPP ACL entries use different TCAM regions. For IPv6 CoPP to work, the IPv6 ACL SUP tcam region (ipv6-sup) needs to be carved to a non- zero size. For more information, see the ACL TCAM Regions, on page 169 and Configuring ACL TCAM Region Sizes, on page 196 topics.

- CoPP can have a maximum of 76 entries for all IPv4 CoPP ACLs, IPv6 CoPP ACLs, and ARP ACLs. The system is programmed with 72 static entries (20 internal, 43 IPv4 ACL, and 9 IPv6 ACL entries). You can configure the remaining 4 entries. If you want to create more entries, you need to delete any unused static CoPP ACEs, and then create your additional entries.

- Cisco Nexus 3000 Series switches drop all the packets when the tunnel is not configured. The packets are also dropped when the tunnel is configured but the tunnel interface is not configured or the tunnel interface is in shut down state.

  Point to Point tunnel (Source and Destination) – Cisco Nexus 3000 Series switches decapsulate all IP-in-IP packets destined to it when the command **feature tunnel** is configured and there is an operational tunnel interface configured with the tunnel source and the destination address that matches the incoming packets' outer source and destination addresses. If there is not a source and destination packet match or if the interface is in shutdown state, the packet is dropped.

  Decapsulate Tunnel (Source only) - Cisco Nexus 3000 Series switches decapsulate all IP-in-IP packets destined to it when the command **feature tunnel** is configured and there is an operational tunnel interface configured with the tunnel source address that matches the incoming packets' outer destination addresses. If there is not a source packet match or if the interface is in shutdown state, the packet is dropped.

- If you use NXAPI over the front panel port, then you must increase the CoPP policy (for http) to allows 3000 PPS traffic, so that there is no packet drop, and the CLIs with a larger output return within the expected time.

- The following limitations apply to Cisco Nexus 34180YC platform:

  - The BC value in the CoPP policy is ignored

  - The transmitted packets to the supervisor moduel are double if the control traffic comes on two pipelines.

  - CIR is in PPS.

  - Default Layer 2 CoPP policies are not supported.

  - Only TTL=1 is supported.

Upgrade Guidelines for CoPP

CoPP has the following upgrade guidelines:

- If you upgrade from a Cisco NX-OS release that does not support the CoPP feature to a release that supports the CoPP feature, CoPP is automatically enabled with the default policy when the switch boots up. You must run the setup script after the upgrade to enable a different policy (default, l3, ,l2). Not configuring CoPP protection can leave your NX-OS device vulnerable to DoS attacks.

- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with additional classes for new protocols, you must run the setup utility for the new CoPP classes to be available.

- Because the setup script modifies the policing rates corresponding to different flows coming into the CPU, we recommend that you run the setup script during a scheduled maintenance period and not during a time when there is traffic on the device.

- When upgrading from Cisco NX-OS Release 6.x to Cisco NX-OS Release 7.x or 9.2x/9.3x, the default control plane policy may not be applied. To apply CoPP policy, you must perform the following steps:

  1. Back up the configuration on Cisco NX-OS Release 6.x

  2. Write erase the switch

  3. Apply the back up configuration

  4. Proceed with the Cisco NX-OS Release 7.x or 7.x or 9.2x/9.3x upgrade

Configuring CoPP

Configuring a Control Plane Class Map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

You can configure policies for IPv4 or IPv6 packets.

Before you begin

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map type control-plane match-any** *class-map-name*
	
	Example:
	
	`switch(config)# class-map type`
	`control-plane ClassMapA`
	`switch(config-cmap)#`
		Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.
		
		Note You cannot use class-default, match-all, or match-any as class map names.

	Command or Action	**Purpose**
Step 3	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name MyAccessList`	Specifies matching for an IP ACL. You can repeat this step to match more than one IP ACL. **Note** The **permit** and **deny** ACL keywords are ignored in the CoPP matching.

Step 4	**exit**
	
	Example:
	`switch(config-cmap)# exit`
	`switch(config)#`
		Exits class map configuration mode.

Step 5	(Optional) **show class-map type control-plane** [*class-map-name*] **Example:** `switch(config)# show class-map type control-plane`	Displays the control plane class map configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the default PPS for that class is 0.

You can configure policies for IPv4 or IPv6 packets.

Before you begin

Ensure that you have configured a control plane class map.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**policy-map type control-plane** *policy-map-name* **Example:**	Specifies a control plane policy map and enters the policy map configuration mode. The policy map name is case sensitive.

	Command or Action	Purpose
	switch(config)# policy-map type control-plane copp-system-policy switch(config-pmap)#	**Note** The name of the policy-map cannot be changed. You can only use the **copp-system-policy** name for the policy-map. The system allows only a single **type control-plane** policy-map to be configured.
Step 3	**class** {*class-map-name* [**insert-before** *class-map-name2*]	**class**} **Example:** switch(config-pmap)# class ClassMapA switch(config-pmap-c)#	Specifies a control plane class map name or the class default and enters control plane class configuration mode.
Step 4	**police** [**pps**] {*pps-value*} [**bc**] *burst-size* [**bytes**	**kbytes**	**mbytes**	**ms**	**packets**	**us**] **Example:** switch(config-pmap-c)# police pps 100 bc 10	Specifies the rate limit in terms of packets per second (PPS) and the committed burst (BC). The PPS range is 0 - 20,000. The default PPS is 0. The BC range is from 0 to 512000000. The default BC size unit is bytes.
Step 5	**police** [**cir**] {*cir-rate* [*rate-type*]} OR **police** [**cir**] {*cir-rate* [*rate-type*]}[**bc**] burst-size [burst-size-type] OR **police** [**cir**] {*cir-rate* [*rate-type*]} **conform transmit** [**violate drop**] **Example:** switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes	You can specify the BC and conform action for the same CIR. The conform action options are as follows: • drop—Drops the packet. • transmit—Transmits the packet.
Step 6	**exit** **Example:** switch(config-pmap-c)# exit switch(config-pmap)#	Exits policy map class configuration mode.
Step 7	**exit** **Example:** switch(config-pmap)# exit switch(config)#	Exits policy map configuration mode.
Step 8	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:** switch(config)# show policy-map type control-plane	Displays the control plane policy map configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring the Control Plane Service Policy

Before you begin

Configure a control plane policy map.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**control-plane**
	
	Example:
	`switch(config) # control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	[**no**] **service-policy input** *policy-map-name*
	
	Example:
	`switch(config-cp)# service-policy input`
	`copp-system-policy`
		Specifies a policy map for the input traffic.

Step 4	**exit**
	
	Example:
	`switch(config-cp)# exit`
	`switch(config)#`
		Exits control plane configuration mode.

Step 5	(Optional) **show running-config copp** [**all**] **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

CoPP Show Commands

To display CoPP configuration information, enter one of the following show commands:

Command	**Purpose**
show ip access-lists [*acl-name*]	Displays all IPv4 ACLs configured in the system, including the CoPP ACLs.

Command	Purpose
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.
show ipv6 access-lists	Displays all of the IPv6 ACLs configured on the device, including the CoPP IPv6 ACLs.
show arp access-lists	Displays all of the ARP ACLs configured on the device, including the CoPP ARP ACLs.
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps and PPS values.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config copp [**all**]	Displays the CoPP configuration in the startup configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Displaying the CoPP Configuration Status

Procedure

	Command or Action	Purpose
Step 1	switch# **show copp status**	Displays the configuration status for the CoPP feature.

Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

Monitoring CoPP

Procedure

	Command or Action	**Purpose**
Step 1	switch# **show policy-map interface control-plane**	Displays packet-level statistics for all classes that are part of the applied CoPP policy.
		Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting).

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

service-policy input: copp-system-policy-default

class-map copp-system-class-igmp (match-any)
match protocol igmp
police cir 1024 kbps , bc 65535 bytes
conformed 0 bytes; action: transmit
violated 0 bytes;
class-map copp-system-class-pim-hello (match-any)
match protocol pim
police cir 1024 kbps , bc 4800000 bytes
conformed 0 bytes; action: transmit
violated 0 bytes;
....


switch# show policy-map interface control-plane
Control Plane

service-policy input: copp-system-policy
class-map copp-s-selfIp (match-any)
 police pps 500
 OutPackets 268
 DropPackets 0
```

Disabling and Reenabling the Rate Limit on CoPP Classes

To transfer data at speeds higher than what is regulated by CoPP, you can disable the default rate limit on CoPP classes and set the rate to the maximum value allowed on the device. Although the packets are now directed to the CPU at the maximum possible rate, the rate of processing of these packets depends on the CPU capability. After data transfer, you must ensure that you reenable the rate limit on CoPP classes.

☞

Important   Disabling the rate limit on CoPP classes can make the CPU vulnerable to overwhelming traffic.

Before you begin

Ensure that the CPU is protected and that excessive external traffic is not directed at device interfaces, the supervisor module or the CPU.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**copp rate-limit disable** **Example:** `switch(config)# copp rate-limit disable`	Disables the default packets per second sent to the CPU and allows the maximum possible packet rate to the CPU on each queue. **Important** After you run this command, a warning appears to notify you that the CoPP rate-limit is disabled for all classes. Hence, the CPU is vulnerable to traffic attacks. Run the **no copp rate-limit disable** command as soon as possible.
Step 3	(Optional) **show policy-map interface control-plane** **Example:** `switch(config)# show policy-map interface control-plane`	Displays packet-level statistics for all classes that are part of the applied CoPP policy. Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting).
Step 4	**no copp rate-limit disable** **Example:** `switch(config)# no copp rate-limit disable`	Resets the rate limit of the packets sent to the CPU on each queue to the default value.

	Command or Action	Purpose
Step 5	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode.

Clearing the CoPP Statistics

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

CoPP Configuration Examples

Creating an IP ACL

```
ip access-list  copp-sample-acl
permit udp any any eq 3333
permit udp any any eq 4444
```

The following example shows how to modify the CoPP Policy to drop all IP-in-IP (Protocol 4) packets immediately if there is not an operational tunnel that matches the incoming packet. Create copp-s-ipinip before the default copp-s-selfip policy as displayed in the following example.

```
ip access-list copp-s-ipinip
10 permit 4 any any
class-map type control-plane match-any copp-s-ipinip
match access-group name copp-s-ipinip
policy-map type control-plane copp-system-policy
class copp-s-ipinip
police pps 0
class copp-s-selfIp
police pps 500
class copp-s-default
police pps 400
```

Creating a Sample CoPP Class with an Associated IP ACL

The following example shows how to create a new CoPP class and associated ACL:

```
class-map type control-plane copp-sample-class
match access-group name copp-sample-acl
```

The following example shows how to add a class to a CoPP policy:

```
policy-map type control-plane copp-system-policy
Class copp-sample-class
 Police pps 100
```

The following example shows how to modify the PPS for an existing class (copp-s-bpdu):

```
policy-map type control-plane copp-system-policy
 Class copp-s-bpdu
 Police pps <new_pps_value>
```

Creating a Dynamic Class (IPv6 ACL)

The following example shows how to create an IPv6 ACL

```
ipv6 access-list  copp-system-acl-eigrp6
10 permit 88 any ff02::a/128
```

Associating an ACL with an Existing or New CoPP Class

The following example shows how to associate an ACL with an existing or new CoPP class:

```
class-map type control-plane copp-s-eigrp
match access-grp name copp-system-acl-eigrp6
```

Adding a Class to a CoPP Policy

The following example shows how to add a class to a CoPP policy, if the class has not already been added:

```
policy-map type control-plane copp-system-policy
class copp-s-eigrp
police pps 100
```

Creating an ARP ACL-Based Dynamic Class

ARP ACLs use ARP TCAM. The default size of this TCAM is 0. Before ARP ACLs can be used with CoPP, this TCAM needs to be carved for a non-zero size.

```
hardware profile tcam region arpacl 128
copy running-config startup-config
reload
```

Creating an ARP ACL

```
arp access-list copp-arp-acl
permit ip 20.1.1.1 255.255.255.0 mac any
```

The procedure to associate an ARP ACLs with a class, and adding that class to the CoPP policy, is the same as the procedure for IP ACLs.

Creating a CoPP Class and Associating an ARP ACL

```
class-map type control-plane copp-sample-class
match access-group name copp-arp-acl
```

Removing a Class from a CoPP Policy

```
policy-map type control-plane copp-system-policy
   no class-abc
```

Removing a Class from the System

```
no class-map type control-plane copp-abc
```

Using the insert-before option to see if a packet matches multiple classes and the priority needs to be assigned to one of them

```
policy-map type control-plan copp-system-policy
class copp-ping insert-before copp-icmp
```

Sample CoPP Configuration

The following example shows a sample CoPP configuration with ACLs, classes, policies, and individual class policing:

```
IP access list copp-system-acl-eigrp
        10 permit eigrp any 224.0.0.10/32
IP access list copp-system-acl-icmp
        10 permit icmp any any
IP access list copp-system-acl-igmp
        10 permit igmp any any
IP access list copp-system-acl-ntp
        10 permit udp any any eq ntp
        20 permit udp any eq ntp any
IP access list copp-system-acl-pimreg
        10 permit pim any any
IP access list copp-system-acl-ping
        10 permit icmp any any echo
        20 permit icmp any any echo-reply
IP access list copp-system-acl-routingproto1
        10 permit tcp any gt 1024 any eq bgp
        20 permit tcp any eq bgp any gt 1024
        30 permit udp any 224.0.0.0/24 eq rip
        40 permit tcp any gt 1024 any eq 639
        50 permit tcp any eq 639 any gt 1024
        70 permit ospf any any
        80 permit ospf any 224.0.0.5/32
        90 permit ospf any 224.0.0.6/32
IP access list copp-system-acl-routingproto2
        10 permit udp any 224.0.0.0/24 eq 1985
        20 permit 112 any 224.0.0.0/24
IP access list copp-system-acl-snmp
        10 permit udp any any eq snmp
        20 permit udp any any eq snmptrap
IP access list copp-system-acl-ssh
        10 permit tcp any any eq 22
        20 permit tcp any eq 22 any
IP access list copp-system-acl-stftp
        10 permit udp any any eq tftp
        20 permit udp any any eq 1758
        30 permit udp any eq tftp any
        40 permit udp any eq 1758 any
        50 permit tcp any any eq 115
        60 permit tcp any eq 115 any
IP access list copp-system-acl-tacacsradius
        10 permit tcp any any eq tacacs
```

```
              20 permit tcp any eq tacacs any
              30 permit udp any any eq 1812
              40 permit udp any any eq 1813
              50 permit udp any any eq 1645
              60 permit udp any any eq 1646
              70 permit udp any eq 1812 any
              80 permit udp any eq 1813 any
              90 permit udp any eq 1645 any
              100 permit udp any eq 1646 any
IP access list copp-system-acl-telnet
              10 permit tcp any any eq telnet
              20 permit tcp any any eq 107
              30 permit tcp any eq telnet any
              40 permit tcp any eq 107 any
IP access list copp-system-dhcp-relay
              10 permit udp any eq bootps any eq bootps
IP access list test
              statistics per-entry
              10 permit ip 1.2.3.4/32 5.6.7.8/32 [match=0]
              20 permit udp 11.22.33.44/32 any [match=0]
              30 deny udp 1.1.1.1/32 any [match=0]


IPv6 access list copp-system-acl-dhcpc6
              10 permit udp any any eq 546
IPv6 access list copp-system-acl-dhcps6
              10 permit udp any ff02::1:2/128 eq 547
              20 permit udp any ff05::1:3/128 eq 547
IPv6 access list copp-system-acl-eigrp6
              10 permit 88 any ff02::a/128
IPv6 access list copp-system-acl-v6routingProto2
              10 permit udp any ff02::66/128 eq 2029
              20 permit udp any ff02::fb/128 eq 5353
IPv6 access list copp-system-acl-v6routingproto1
              10 permit 89 any ff02::5/128
              20 permit 89 any ff02::6/128
              30 permit udp any ff02::9/128 eq 521


class-map type control-plane match-any copp-icmp
  match access-group name copp-system-acl-icmp
class-map type control-plane match-any copp-ntp
  match access-group name copp-system-acl-ntp
class-map type control-plane match-any copp-s-arp
class-map type control-plane match-any copp-s-bfd
class-map type control-plane match-any copp-s-bpdu
class-map type control-plane match-any copp-s-dai
class-map type control-plane match-any copp-s-default
class-map type control-plane match-any copp-s-dhcpreq
  match access-group name copp-system-acl-dhcps6
class-map type control-plane match-any copp-s-dhcpresp
  match access-group name copp-system-acl-dhcpc6
  match access-group name copp-system-dhcp-relay
class-map type control-plane match-any copp-s-eigrp
  match access-group name copp-system-acl-eigrp
  match access-group name copp-system-acl-eigrp6
class-map type control-plane match-any copp-s-glean
class-map type control-plane match-any copp-s-igmp
  match access-group name copp-system-acl-igmp
class-map type control-plane match-any copp-s-ipmcmiss
class-map type control-plane match-any copp-s-l2switched
class-map type control-plane match-any copp-s-l3destmiss
class-map type control-plane match-any copp-s-l3mtufail
class-map type control-plane match-any copp-s-l3slowpath
class-map type control-plane match-any copp-s-pimautorp
class-map type control-plane match-any copp-s-pimreg
```

```
      match access-group name copp-system-acl-pimreg
class-map type control-plane match-any copp-s-ping
  match access-group name copp-system-acl-ping
class-map type control-plane match-any copp-s-ptp
class-map type control-plane match-any copp-s-routingProto1
  match access-group name copp-system-acl-routingproto1
  match access-group name copp-system-acl-v6routingproto1
class-map type control-plane match-any copp-s-routingProto2
  match access-group name copp-system-acl-routingproto2
class-map type control-plane match-any copp-s-selfIp
class-map type control-plane match-any copp-s-ttl1
class-map type control-plane match-any copp-s-v6routingProto2
  match access-group name copp-system-acl-v6routingProto2
class-map type control-plane match-any copp-snmp
  match access-group name copp-system-acl-snmp
class-map type control-plane match-any copp-ssh
  match access-group name copp-system-acl-ssh
class-map type control-plane match-any copp-stftp
  match access-group name copp-system-acl-stftp
class-map type control-plane match-any copp-tacacsradius
  match access-group name copp-system-acl-tacacsradius
class-map type control-plane match-any copp-telnet
  match access-group name copp-system-acl-telnet
policy-map type control-plane copp-system-policy
  class copp-s-selfIp
    police pps 500
  class copp-s-default
    police pps 400
  class copp-s-l2switched
    police pps 200
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-l3slowpath
    police pps 100
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-dai
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 1300
  class copp-s-v6routingProto2
    police pps 1300
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
  class copp-s-pimautorp
    police pps 200
  class copp-s-routingProto1
    police pps 1000
```

```
      class copp-s-arp
        police pps 200
      class copp-s-ptp
        police pps 1000
      class copp-s-bfd
        police pps 350
      class copp-s-bpdu
        police pps 12000
      class copp-icmp
        police pps 200
      class copp-telnet
        police pps 500
      class copp-ssh
        police pps 500
      class copp-snmp
        police pps 500
      class copp-ntp
        police pps 100
      class copp-tacacsradius
        police pps 400
      class copp-stftp
        police pps 400
    control-plane
      service-policy input copp-system-policy
```

Example: Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility:

```
switch# setup

        ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n

  Enter the switch name : switch

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway for mgmt? (yes/no) [y]: n

  Enable the telnet service? (yes/no) [n]: y
```

```
   Enable the ssh service? (yes/no) [y]: n

   Configure the ntp server? (yes/no) [n]: n

   Configure CoPP System Policy Profile ( default / l2 / l3 ) [default]: l2

The following configuration will be applied:
  switchname switch
  telnet server enable
  no ssh server enable
  policy-map type control-plane copp-system-policy ( l2 )

Would you like to edit the configuration? (yes/no) [n]: n

Use this configuration and save it? (yes/no) [y]: y

[########################################] 100%
```

Preventing CoPP Overflow by Splitting ICMP Pings

Note   This section applies only to Cisco Nexus 3000 Series switches and Cisco Nexus 3100 Series switches in N3K mode.

Some servers use ICMP pings to the default gateway to verify that the active NIC still has access to the aggregation switch. As a result, if the CoPP values are exceeded, CoPP starts dropping traffic for all networks. One malfunctioning server can send out thousands of ICMP pings, causing all servers in one aggregation block to lose their active NIC and start swapping NICs.

If your server is configured as such, you can minimize the CoPP overflow by splitting the ICMP pings based on subnets or groups of subnets. Then if a server malfunctions and overflows CoPP, the supervisor answers the ICMP pings only on some subnetworks.

The last entry in the class map or policy map should identify all of the ICMP pings in the networks that are not specified. If these counters increase, it means that a new network was added that was not specified in the existing ACLs for ICMP. In this case, you would need to update the ACLs related to ICMP.

Note   Per the default CoPP, ICMP pings fall under copp-system-p-class-monitoring.

The following example shows how to prevent CoPP overflow by splitting ICMP pings.

First, add the new ACLs that identify the networks you want to group together based on the findings of the investigations of the applications:

```
ip access-list copp-icmp-1
statistics per-entry
10 permit icmp 10.2.1.0 255.255.255.0 any
20 permit icmp 10.2.2.0 255.255.255.0 any
30 permit icmp 10.2.3.0 255.255.255.0 any
ip access-list copp-icmp-2
statistics per-entry
10 permit icmp 10.3.1.0 255.255.255.0 any
10 permit icmp 10.3.2.0 255.255.255.0 any
```

```
10 permit icmp 10.3.3.0 255.255.255.0 any
ip access-list copp-icmp-3
statistics per-entry
10 permit icmp 10.4.1.0 255.255.255.0 any
10 permit icmp 10.4.2.0 255.255.255.0 any
10 permit icmp 10.4.3.0 255.255.255.0 any
...
ip access-list copp-icmp-10
10 permit icmp any any
```

Add these ACLs to the new class maps for CoPP:

```
class-map type control-plane match-any copp-cm-icmp-1
 match access-group name copp-icmp-1
class-map type control-plane match-any copp-cm-icmp-2
 match access-group name copp-icmp-2
class-map type control-plane match-any copp-cm-icmp-3
 match access-group name copp-icmp-3
...
class-map type control-plane match-any copp-cm-icmp-10
 match access-group name copp-icmp-10
```

Modify the CoPP policy map by adding new policies with the above created class maps:

```
policy-map type control-plane copp-system-p-policy
class copp-cm-icmp-1
     police cir X pps bc X conform transmit violate drop
class copp-cm-icmp-2
     police cir X pps bc X conform transmit violate drop
class copp-cm-icmp-3
     police cir X pps bc X conform transmit violate drop
class copp-cm-icmp-4
     police cir X pps bc X conform transmit violate drop
class copp-cm-icmp-10
     police cir X pps bc X conform transmit violate drop
```

Delete ICMP from the existing class maps:

```
class-map type control-plane match-any copp-system-p-class-monitoring
no match access-grp name copp-system-p-acl-icmp
```

Additional References for CoPP

This section provides additional information related to implementing CoPP.

Related Documents

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	

Configuring Rate Limits

This chapter contains the following sections:

About Rate Limits

You can configure span-egress hardware rate-limit to restrict amount of ERSPAN monitor traffic transmitted out of the Cisco NX-OS device. You can configure rate limits in packets per second for the following types of traffic:

- SPAN egress traffic—For this option, you can configure rate limits in kilobits per seconds.

Licensing Requirements for Rate Limits

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	No license is required for rate limits. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for Rate Limits

Rate limits has the following configuration guidelines and limitations:

- You can configure a hardware rate-limiter to show statistics for outbound ERSPAN monitor traffic on egress ports.

    The rate-limiter on egress ports is limited per ASIC, rather than per port or SPAN session.

    The rate-limiter only applies to ERSPAN and not local SPAN traffic.

    sFlow and ERSPAN cannot co-exist in the same Cisco Nexus NFE2-enabled devices.

Note   If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for Rate Limits

This table lists the default settings for rate limits parameter.

Table 21: Default Rate Limits Parameters Settings

Parameters	Default
SPAN egress rate limit	No limits

Configuring Rate Limits

You can set rate limits on supervisor-bound traffic.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hardware rate-limiter span-egress** *rate*
	
	Example:
	
	`switch(config)# hardware rate-limiter`
	`span-egress 100`
		Configures rate limits in kilobits per second for SPAN for egress traffic. The range is from 0 to 100000000.

	Command or Action	Purpose
		Note You should not configure both sFlow and the SPAN egress rate-limiter because the SPAN egress rate-limiter can affect the functionality of sFLOW.
Step 3	(Optional) **show hardware rate-limiter span-egress** **Example:** `switch# show hardware rate-limiter span-egress`	Displays the rate limit configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Monitoring Rate Limits

You can monitor rate limits.

Procedure

	Command or Action	Purpose
Step 1	**show hardware rate-limiter span-egress** **Example:** `switch# show hardware rate-limiter span-egress`	Displays the rate limit statistics.

Clearing the Rate Limit Statistics

You can clear the rate limit statistics.

Procedure

	Command or Action	Purpose
Step 1	**clear hardware rate-limiter span-egress** **Example:** `switch# clear hardware rate-limiter span-egress`	Clears the rate limit statistics.

Verifying the Rate Limit Configuration

To display the rate limit configuration information, perform the following tasks:

Command	Purpose
show hardware rate-limiter | **span-egress**	Displays the rate limit configuration.

Configuration Examples for Rate Limits

The following shows an example of the rate-limiter configuration for ERSPAN:

```
switch(config)# hardware rate-limiter span-egress 100
Warning: This span-egress rate-limiter might affect functionality of sFlow
switch(config)# show hardware rate-limiter span-egress
Units for Config: packets per second (kilo bits per second for span-egress)
Allowed, Dropped & Total: aggregated since Module: 1
R-L Class        Config          Allowed      Dropped       Total
+---------------+----------+-------------+-------------+---------------+
   span-egress          123        0             0             0
<<configured
```

Additional References for Rate Limits

This section includes additional information related to implementing rate limits.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*

INDEX