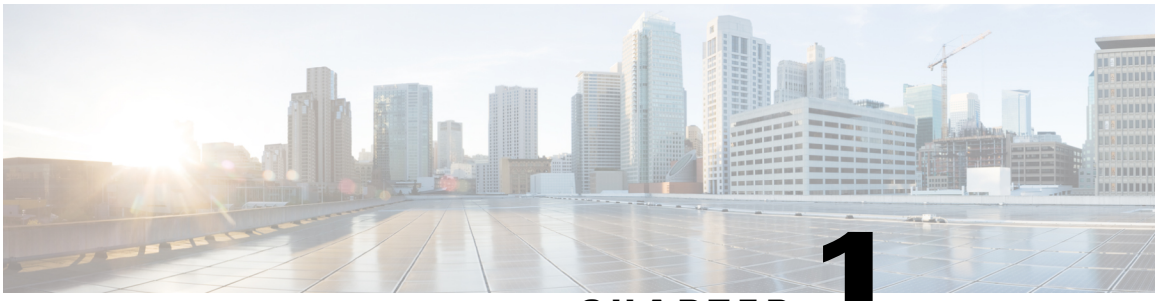
Cisco Nexus 1000V for KVM Security Configuration Guide, Release 5.x

First Published: August 01, 2014

Last Modified: November 13, 2015

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

CONTENTS



CHAPTER 1

New and Changed Information

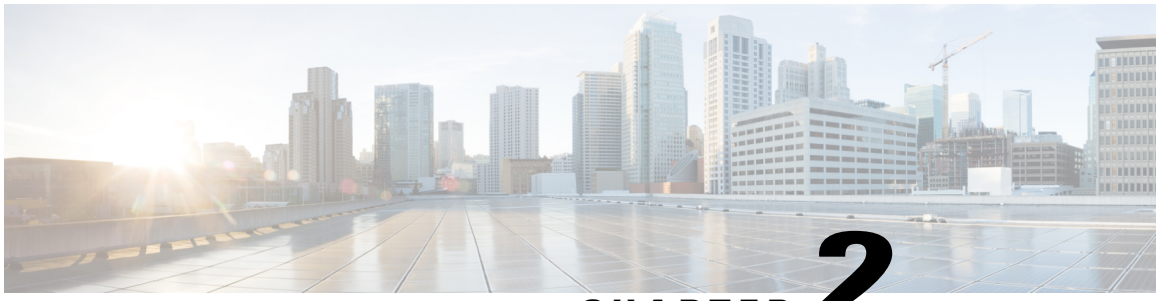
This chapter lists new and changed content in this document by software release.

New and Changed Information

Table 1: New and Changed Features

Content	Description	Changed in Release	Where Documented
Unknown Unicast Packet Flooding (UUFB)	This feature is introduced.	5.2(1)SK3(2.1)	Blocking Unknown Unicast Flooding, on page 107

New and Changed Information



CHAPTER 2

Overview

This chapter contains the following sections:

Cisco Nexus 1000V for KVM and OpenStack

The Cisco Nexus 1000V for KVM consists of two main components:

- Virtual Ethernet Module (VEM)—A software component that is deployed on each kernel-based virtual machine (VM) host. Each VM on the host is connected to the VEM through virtual Ethernet (vEth) ports.
- Virtual Supervisor Module (VSM)—The Management component that controls multiple VEMs and helps in the definition of VM-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the Cisco Cloud Services Platform appliance.

Each of these components is tightly integrated with the OpenStack environment:

- The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.
- The VSM is integrated with OpenStack using the OpenStack Neutron Plug-in.
- The OpenStack Neutron API has been extended to include two additional user-defined resources:
  - Network profiles are logical groupings of network segments.
  - Policy profiles group port policy information, including security.

Using OpenStack, you create VMs, networks, and subnets on the Cisco Nexus 1000V for KVM, by defining components such as the following:

- Tenants
- Network segments, such as VLANs, VLAN trunks, and VXLANs
- IP address pools (subnets)

Using the Cisco Nexus 1000V for KVM VSM, you create port profiles (called policy profiles in OpenStack), which define the port policy information, including security settings.

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for KVM for each unique combination of policy port profile and network segment. All other VMs deployed with the same policy to this network reuse this dynamic port profile.

Note You must consistently use OpenStack for all VM network and subnet configuration. If you use *both* OpenStack and the VSM to configure VM networks and subnets, the OpenStack and the VSM configurations can become out-of-sync and result in faulty or inoperable network deployments.

Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) is an architectural framework for configuring a set of three independent, consistent, and modular security functions

- 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.

RADIUS Security Protocol

AAA establishes communication between your network access server and your RADIUS security server. RADIUS is 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+ Security Protocol

AAA establishes communication between your network access server and your TACACS+ security server.

TACACS+ is 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 that usually runs on a UNIX or Windows NT workstation. TACACS+ provides separate and modular authentication, authorization, and accounting facilities.

SSH

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

The SSH client works with publicly and commercially available SSH servers.

Telnet

You can use the Telnet protocol to set up TCP/IP connections to a host. Telnet allows a person 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.

Access Control Lists

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.

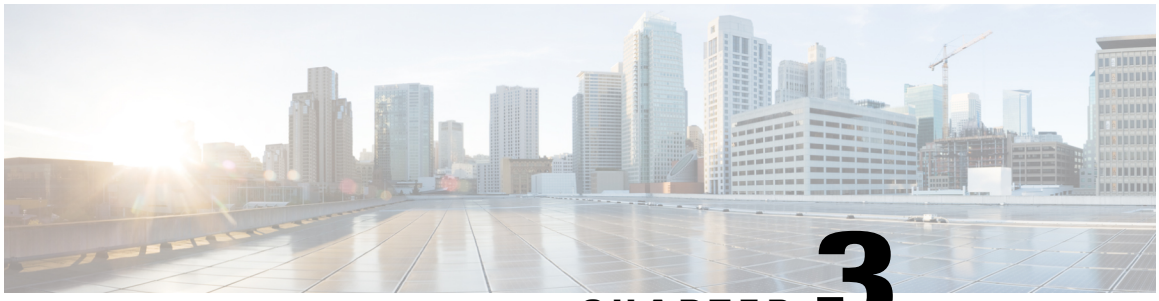
MAC ACLs

MAC ACLs are ACLs that filter traffic using the information in the Layer 2 header of each packet. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that a MAC 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.

Unknown Unicast Flooding

Unknown unicast packet flooding (UUFB) limits unknown unicast flooding in the forwarding path to prevent the security risk of unwanted traffic reaching the Virtual Machines (VMs). UUFB prevents packets received on both vEthernet and Ethernet interfaces destined to unknown unicast addresses from flooding the VLAN. When UUFB is applied, Virtual Ethernet Modules (VEMs) drop unknown unicast packets received on uplink ports, while unknown unicast packets received on vEthernet interfaces are sent out only on uplink ports.



CHAPTER 3

Configuring AAA

This chapter contains the following sections:

Information about AAA

AAA Security Services

Based on a user ID and password combination, authentication, authorization, and accounting (AAA) is used to authenticate and authorize users. A key secures communication with AAA servers.

In many circumstances, AAA uses protocols such as RADIUS or TACACS+ 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.

Although AAA is the primary (and recommended) method for access control, additional features for simple access control are available outside the scope of AAA, such as local username authentication, line password authentication, and enable password authentication. However, these features do not provide the same degree of access control that is possible by using AAA.

Separate AAA configurations are made for the following services:

- User Telnet or Secure Shell (SSH) login authentication

- Console login authentication
- User management session accounting

The following table provides the authentication commands:

AAA Service Configuration Option	Related Command
Telnet or SSH login	aaa authentication login default
Console login	aaa authentication login console

Authentication

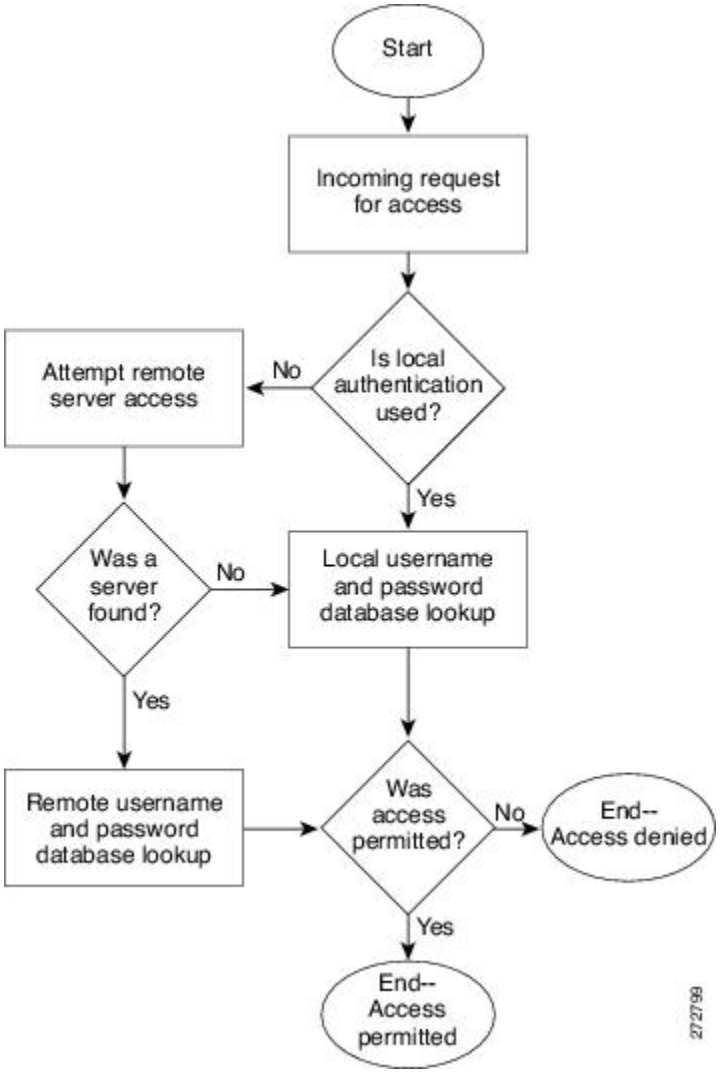
Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and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

Authentication is accomplished as follows:

Authentication Method	Description
Local database	Authenticates the following with a local lookup database of usernames or passwords: • Console login authentication • User login authentication • User management session accounting
Remote RADIUS or TACACS+ server	Authenticates the following with a local lookup database of usernames or passwords: • Console login authentication • User login authentication • User management session accounting
None	Authenticates the following with only a username: • Console login authentication • User login authentication • User management session accounting

The following figure shows a flowchart of the authentication process.

Figure 1: Authenticating User Login



Note This diagram 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.

Authorization

Authorization restricts the actions that a user is allowed to perform. It 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

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.

Accounting tracks and maintains a log of every SVS management session. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

AAA Server Groups

Remote AAA server groups can provide failovers if one remote AAA server fails to respond, which means that if the first server in the group fails, the next server in the group is tried until a server responds. Multiple server groups can provide failovers for each other in this same way.

If all remote server groups fail, the local database is used for authentication.

Prerequisites for AAA

- At least one TACACS+ or RADIUS server is IP reachable
- The Virtual Supervisor Module (VSM) is configured as an AAA server client.
- A shared secret key is configured on the VSM and the remote AAA server.

Guidelines and Limitations

The Cisco Nexus 1000V does not support usernames that have all numeric characters and does not create local usernames that have all numeric characters. If a username that has all numeric characters already exists on an AAA server and is entered during login, the Cisco Nexus 1000V does not authenticate the user.

AAA Default Settings

Parameters	Default
Console authentication method	local
Default authentication method	local

Parameters	Default
Login authentication failure messages	Disabled

Configuring AAA

Configuring a Login Authentication Method

Before You Begin

Log in to the CLI in EXEC mode.

If authenticate with TACACS+ server groups, you must have already added the groups.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login** {**console** | **default**} {**group** *group-list* [**none**] | **local** | **none**}
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login** {**console** | **default**} {**group** *group-list* [**none**] | **local** | **none**}	Configures the console or default login authentication method. the keywords and arguments are as follows: • console—??? • default—??? • group—Specifies that authentication is done by server groups. • *group-list*—List of server group names separated by spaces; or none for no authentication. • none— Specifies no authentication. • local—Specifies that the local database is used for authentication. **Note** Local is the default and is used when no methods are configured or when all the configured methods fail to respond. • none—Specifies that authentication is done by the username.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch# **show aaa authentication**	(Optional) Displays the configured login authentication method.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a local authentication method:

```
switch# configure terminal
switch(config)# aaa authentication login console group tacgroup
switch(config)# exit
switch# show aaa authentication
         default: group tacgroup
         console: group tacgroup
switch# copy running-config startup-config
switch#
switch# configure terminal
switch(config)# aaa authentication login default group tacacs
switch(config)# aaa authentication login console group tacacs
```

Enabling Login Authentication Failure Messages

You can enable the login authentication failure message to display if the remote AAA servers do not respond.

The following is the login authentication failure message:

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

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login error-enable**
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication login error-enable**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

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

The following example shows how to enable the login authentication failure message:

```
switch# configure terminal
switch(config)# aaa authentication login error-enable
switch(config)# exit
switch# show aaa authentication login error-enable
enabled
```

Verifying the AAA Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
show aaa groups	Displays the AAA server group configuration.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

Example 1: show aaa authentication

```
switch# show aaa authentication login error-enable
disabled
switch#
```

Example 2: show aaa groups

```
switch# show aaa groups
radius
switch#
```

Example 3: show running config aaa

```
switch# show running-config aaa all
!Time: Fri Nov 15 11:22:13 2013

version 5.2(1)SK1(2.1)
aaa authentication login default local
aaa authorization ssh-publickey default local
aaa authorization ssh-certificate default local
aaa accounting default local
aaa user default-role
aaa authentication login default fallback error local
aaa authentication login console fallback error local
no aaa authentication login error-enable
no aaa authentication login mschap enable
no aaa authentication login mschapv2 enable
no aaa authentication login chap enable
no aaa authentication login ascii-authentication
no radius-server directed-request
switch#
```

Example 4: show startup-config aaa

```
switch# show startup-config aaa
!Command: show startup-config aaa
!Time: Fri Nov 15 11:19:57 2013
!Startup config saved at: Tue Jun 25 05:23:10 2013

version 5.2(1)SK1(2.1)
switch#
```

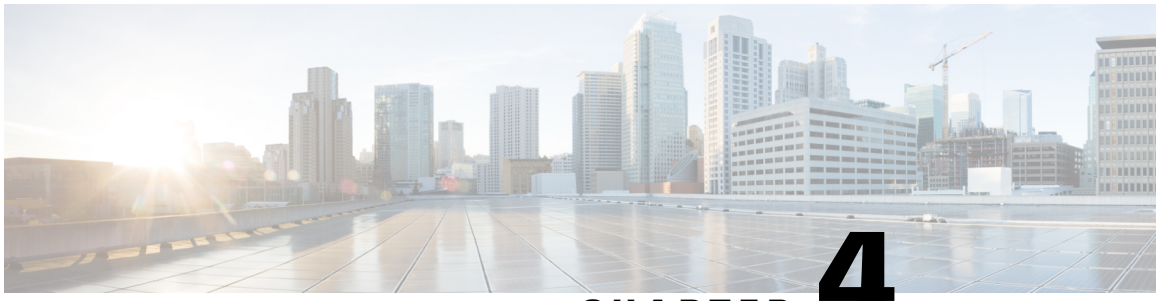
Configuration Examples for AAA

The following example configures AAA:

```
switch# show aaa authentication login default group tacacs
switch# show aaa authentication login console group tacacs
```

Feature History for AAA

Feature Name	Releases	Feature Information
AAA	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 4

Configuring RADIUS

This chapter contains the following sections:

Information About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco NX-OS 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 NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a 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 NX-OS device to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

RADIUS Operation

When a user attempts to log in and authenticate to a Cisco NX-OS device using RADIUS, the following 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:

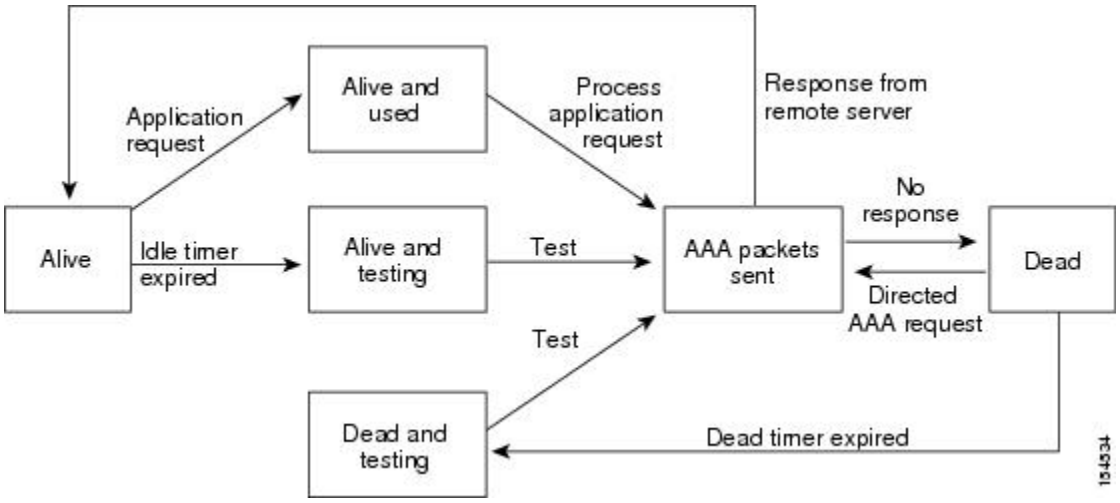
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. Unresponsive RADIUS servers are marked as dead and are not sent AAA requests. Dead RADIUS servers are periodically monitored and returned to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and an error message is displayed indicating that a failure is taking place.

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.

Figure 2: Radius Server States



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 = (equal sign) for mandatory attributes and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication, 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:

- Shell—Protocol used in access-accept packets to provide user profile information.
- Accounting—Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The following attributes are supported:

- 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. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be “network-operator vdc-admin.” This attribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can be used only with the shell protocol value. The following examples show the roles attribute as supported by Cisco Access Control System (ACS):

  ```
  shell:roles=“network-operator vdc-admin”
  ```

  ```
  shell:roles*“network-operator vdc-admin”
  ```

  The following examples show the roles attribute as supported by FreeRADIUS:

  ```
  Cisco-AVPair = ”shell:roles=\“network-operator vdc-admin\””
  ```

  ```
  Cisco-AVPair = “shell:roles*\“network-operator vdc-admin\””
  ```

  If you are using Cisco ACS and intend to use the same ACS group for both Cisco Nexus 1000V and Cisco UCS authentication, use the following roles attribute:

  ```
  cisco-av-pair*shell:roles="network-admin admin"
  ```

  Note When you specify a VSA as shell:roles*"network-operator vdc-admin" or "shell:roles*\"network-operator vdc-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- 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).

Prerequisites for RADIUS

- You already know the RADIUS server IP addresses or hostnames.
- You already know the key(s) used to secure RADIUS communication in your network.
- The device is already configured as a RADIUS client of the AAA servers.

Guidelines and Limitations

You can configure a maximum of 64 RADIUS servers.

Default Settings

Table 2: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting

Parameters	Default
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

Configuring RADIUS Server Hosts

You can configure the IP address or the hostname for each RADIUS server to be used for authentication. You should know the following information:

- You can configure up to 64 RADIUS servers.
- All RADIUS server hosts are automatically added to the default RADIUS server group.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *hostname*}
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *hostname*}	Defines the IP address or hostname for the RADIUS server or the RADIUS server Domain Name Server (DNS) name. *ipv4-address*—The IP address for the RADIUS server.

	Command or Action	Purpose
		hostname—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures a RADIUS server host:

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

Configuring the Global RADIUS Key

You can configure the key that is used by all RADIUS servers to authenticate with the Cisco Nexus 1000V.

You must know the global key that is used for RADIUS server authentication.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server key** [**0** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	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 cleartext (0) or encrypted (7) preshared key. The default format is cleartext. *key-value*—The preshared key value. The maximum length is 63 characters. By default, no preshared key is configured.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS 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	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The follow example configures the global RADIUS key:

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

Configuring a RADIUS Server Key

You can configure a key for a single RADIUS server host.

You must have the key for the remote RADIUS host.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific RADIUS server. You can specify a cleartext (**0**) or encrypted (**7**) preshared key. The default format is cleartext. *ipv4-address*—The IP address for the RADIUS server.

	Command or Action	Purpose
		hostname—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *key-value*—The preshared key value. The maximum length is 63 characters.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS 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	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures a RADIUS server key:

```
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 configure a RADIUS server group whose member servers share authentication functions.

The servers in the group are tried in the same order in which you configure them

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that all servers in a RADIUS server group must belong to the RADIUS protocol.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa group server radius** *group-name*
3. switch(config-radius)# **server** {*ipv4-address* | *server-name*}
4. (Optional) switch(config-radius)# **deadtime** *minutes*
5. (Optional) switch(config-radius)# **use-vrf** *vrf-name*
6. (Optional) switch(config-radius)# **source-interface** {*interface-type*} {*interface-number*}
7. (Optional) switch(config-radius)# **show radius-server groups** [*group-name*]
8. (Optional) switch(config-radius)# **copy running-config startup-config**

DETAILED STEPS

	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 RADIUS server group configuration mode for that group. *group-name*—The name of the server group. The name is a case-sensitive, alphanumeric string with a maximum length of 127 characters.
Step 3	switch(config-radius)# **server** {*ipv4-address* | *server-name*}	Configures the RADIUS server as a member of the RADIUS server group. *ipv4-address*—The IP address for the RADIUS server. *server-name*—The name of the RADIUS server. The name is alphanumeric, case sensitive, and has a maximum of 256 characters. **Tip** If the specified RADIUS server is not found, configure it using the radius-server host command and retry this command.
Step 4	switch(config-radius)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time. *minutes*—The dead time, in minutes. The range is from 1 to 1440. The default value is 0 minutes. **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	switch(config-radius)# **use-vrf** *vrf-name*	(Optional) Specifies the virtual routing and forwarding (VFR) to use to contact the servers in the server group.
Step 6	switch(config-radius)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) Specifies a source interface to be used to reach the RADIUS server. *interface-type*—The interface type. *interface-number*—The interface number. The interface types and interface numbers are defines as follows: • loopback—Virtual interface number from 0 to 1023 • mgmt—Management interface 0 • null—Null interface 0 • port-channel—Port channel number from 1 to 4096
Step 7	switch(config-radius)# **show radius-server groups** [*group-name*]	(Optional) Displays the RADIUS server group configuration. *group-name*—The name of the server group. The name is a case-sensitive, alphanumeric string with a maximum length of 127 characters.

	Command or Action	Purpose
Step 8	switch(config-radius)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

The following example configures 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 vrf1
switch(config-radius)# source-interface mgmt0
switch(config-radius)# show radius-server group
total number of groups:2

following RADIUS server groups are configured:
        group Radserver:
                server: 10.10.1.1
                deadtime is 30
        group test:
                deadtime is 30
switch(config-radius)# copy running-config startup-config
```

Enabling RADIUS Server Directed Requests

You can allow users to designate the RADIUS server to send their authentication request to. This is called a directed request.

If you enable this option, a user can log in as username@vrfname:hostname, where *vrfname* is the virtual routing and forwarding (VRF) instance to use and *hostname* is the name of a configured RADIUS server.

Directed requests are disabled by default.

Note User-specified logins are supported only for Telnet sessions.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server directed-request**
3. switch(config)# **exit**
4. (Optional) switch(config)# **show radius-server directed-request**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server directed-request**	Enables directed requests. The default is disabled.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch(config)# **show radius-server directed-request**	(Optional) Displays the directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example enables RADIUS server directed requests:

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

Setting the Global Timeout for All RADIUS Servers

You can configure the global timeout interval that specifies how long to wait for a response from a RADIUS server before declaring a timeout failure.

The timeout specified in Setting the Timeout Interval for a Single RADIUS Server , on page 27 overrides the global RADIUS timeout.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server timeout** *seconds*
3. switch(config-radius)# **exit**
4. (Optional) switch(config-radius)# **show radius-server**
5. (Optional) switch(config-radius)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server timeout** *seconds*	Specifies the transmission timeout interval for RADIUS servers. *seconds*—The transmission timeout interval, in seconds. The range is from 1 to 60 seconds. The default value is 5 seconds.
Step 3	switch(config-radius)# **exit**	Returns you to the EXEC mode.
Step 4	switch(config-radius)# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch(config-radius)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

The following example sets the global timeout for all RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server timeout 101
switch(config-radius)# exit
switch(config-radius)# show radius-server
switch(config-radius)# copy running-config startup-config
```

Configuring a Global Retry Count for All RADIUS Servers

You can configure the maximum number of times to retry transmitting to a RADIUS server before reverting to local authentication. This setting is applied to all RADIUS servers.

By default, retransmission to a RADIUS server is only tried once before reverting to local authentication. You can increase the number of retries up to a maximum of five. The retry count specified for a single RADIUS server in Configuring Retries for a Single RADIUS Server overrides this global setting.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server retransmit***count*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server retransmit***count*	Defines the number of retransmits allowed before reverting to local authentication. This global setting applies to all RADIUS servers. *count*—The number of allowed retransmits. The range is from 0 to 5. The default value is 1.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures the global retry count for all RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server retransmit 31
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Setting the Timeout Interval for a Single RADIUS Server

You can configure how long to wait for a response from a RADIUS server before declaring a timeout failure. The timeout specified for a single RADIUS server overrides the timeout defined in Setting the Global Timeout for All RADIUS Servers, on page 25.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name* } **timeout** *seconds*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name* } **timeout** *seconds*	Specifies the timeout interval for the specified server. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *seconds*—The timeout interval. The range is from 1 to 60 seconds. The default is 5 seconds. **Note** The timeout specified for a single RADIUS server overrides the global RADIUS timeout.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example sets the timeout interval for a single RADIUS server:

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

Configuring Retries for a Single RADIUS Server

You can configure the maximum number of times to retry transmitting to a RADIUS server before reverting to local authentication. This setting applies to a single RADIUS server and takes precedence over the global retry count.

Before You Begin

Log in to the CLI in EXEC mode.

Know the following:

- By default, retransmission to a RADIUS server is only tried once before reverting to local authentication.
- You can increase the number of retries up to a maximum of five.
- The retry count specified for a single RADIUS server overrides the global setting made for all RADIUS servers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *count*—The retransmission count. The default value is the global value. **Note** This retransmit count for a single RADIUS server overrides the global setting for all RADIUS servers.
Step 3	switch(config)# **exit**	Returns you to EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures retries for a single RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host server1 retransmit 3
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring a RADIUS Accounting Server

You can configure a server to perform accounting functions.

By default, RADIUS servers are used for both accounting and authentication.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the destination UDP port number for RADIUS accounting messages.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **acct-port** *udp-port*
3. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **accounting**
4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **acct-port** *udp-port*	(Optional) Associates a specific host with the UDP port that receives RADIUS accounting messages. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *upd-port*—The UPD port number. The range is from 0 to 65535. The default value is 1812.
Step 3	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **accounting**	(Optional) Designates the specific RADIUS host as an accounting server. The default is both accounting and authentication. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 4	switch(config)# **exit**	Returns you to the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures a RADIUS accounting 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)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring a RADIUS Authentication Server

You can configure a server to perform authentication functions.

By default, RADIUS servers are used for both accounting and authentication.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the destination UDP port number for RADIUS authentication messages.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **auth-port** *udp-port*
3. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **authentication**
4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **auth-port** *udp-port*	(Optional) Associates a specific host with the UDP port that receives RADIUS authentication messages. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *upd-port*—The UPD port number. The range is from 0 to 65535. The default value is 1812.
Step 3	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **authentication**	(Optional) Designates the specific RADIUS host as an authentication server. The default is both accounting and authentication. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 4	switch(config)# **exit**	Returns you to the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration

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

The following example configures a RADIUS authentication server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.2.2 auth-port 2005
switch(config)# radius-server host 10.10.2.2 authentication
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring Periodic RADIUS Server Monitoring

You can configure the monitoring of RADIUS servers.

The test idle timer specifies the interval of time that elapses before a test packet is sent to a unresponsive RADIUS server

The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco NX-OS device does not perform periodic RADIUS server monitoring.

Note For security reasons, do not configure a username that is in the RADIUS database as a test username.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. switch(config)# **radius-server dead-time** *minutes*
4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *hostname*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. *ipv4-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *minutes*—The idle time, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes. *name*—The username to use when connecting to the RADIUS server. The default value is test. *password*—The user's password. The default value is test. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	switch(config)# **radius-server dead-time** *minutes*	Specifies the number of minutes to wait before sending a test packet to a RADIUS server that was declared dead. *minutes*—The amount of time to wait, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes.
Step 4	switch(config)# **exit**	Returns you to the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures periodic RADIUS server monitoring:

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

Configuring the Global Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time to wait after declaring a RADIUS server dead, before sending out a test packet to determine if the server is now alive.

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.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server deadtime** *minutes*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

The following example configures the global dead-time interval:

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

Manually Monitoring RADIUS Servers or Groups

You can manually send a test message to a RADIUS server or to a server group.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **test aaa server radius** {*ipv4-address* | *hostname*} [**vrf** *vrf-name*] *username password*
2. switch# **test aaa group** *group-name username password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **test aaa server radius** {*ipv4-address* | *hostname*} [**vrf** *vrf-name*] *username password*	Sends a test message to a RADIUS server to confirm availability. *ipv4-address*—The IP address of the RADIUS server. *hostname*—The hostname of the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *vrf-name*—The Virtual Routing and Forwarding (VRF) name.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a RADIUS server group to confirm availability. *group-name*—The name of the RADIUS server group. *username*—The username to use when connecting to the RADIUS server group. *password*—The user's password.

The following example manually monitors a RADIUS server and a RADIUS server group:

```
switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Verifying the RADIUS Configuration

Use one of the following commands to verify the configuration:

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 [*hostname* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *ipv4-address*—The IP address for the RADIUS server.

Displaying RADIUS Server Statistics

Use the following command to display statistics for RADIUS server activity:

show radius-server statistics { *hostname* | *ipv4-address* }
hostname—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters.

ipv4-address—The IP address for the RADIUS server.

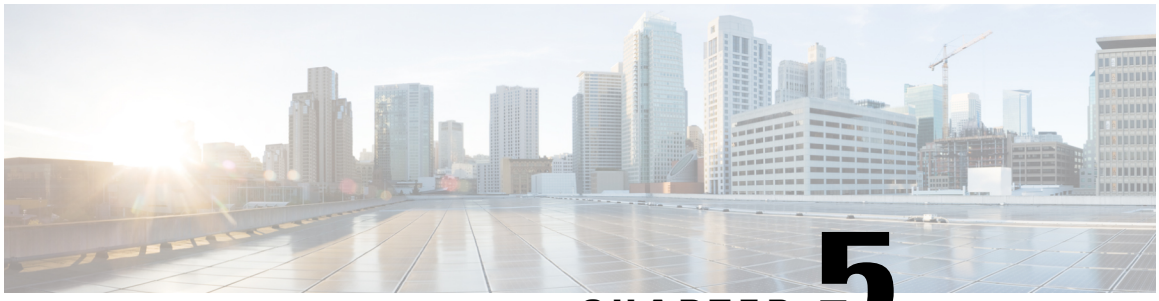
Configuration Example for RADIUS

The following example shows how to configure a global RADIUS key and a RADIUS server host key:

```
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
    server 10.10.1.1
```

Feature History for RADIUS

Feature Name	Releases	Feature Information
RADIUS	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 5

Configuring TACACS+

This chapter contains the following sections:

Information About TACACS+

The TACACS+ security protocol provides centralized validation of users who are attempting to gain access to a device. TACACS+ services are maintained in a database on a TACACS+ daemon that is running, typically, 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 device are available.

TACACS+ provides for separate authentication, authorization, and accounting services. The TACACS+ daemon provides each service 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 TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. Centralized authentication is provided using the TACACS+ protocol.

TACACS+ Operation for User Login

The following sequence of events take place when you attempt to log in to a TACACS+ server using the Password Authentication Protocol (PAP):

1 When a connection is established, the TACACS+ daemon is contacted 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 might include prompts for additional information, such as your mother's maiden name.

2 The TACACS+ daemon provides one of the following responses:

   a ACCEPT—User authentication succeeds and service begins. If user authorization is needed, authorization begins.

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

   c ERROR—An error occurred at some time during authentication either at the daemon or in the network connection. If an ERROR response is received, the device tries to use an alternative method for authenticating the user.

   If further authorization is required after authentication, the user also undergoes an additional authorization phase. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3 If TACACS+ authorization is required, the TACACS+ daemon is contacted 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, access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

You must configure the TACACS+ preshared key to authenticate to the TACACS+ server. A preshared key is a secret text string shared between the 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.

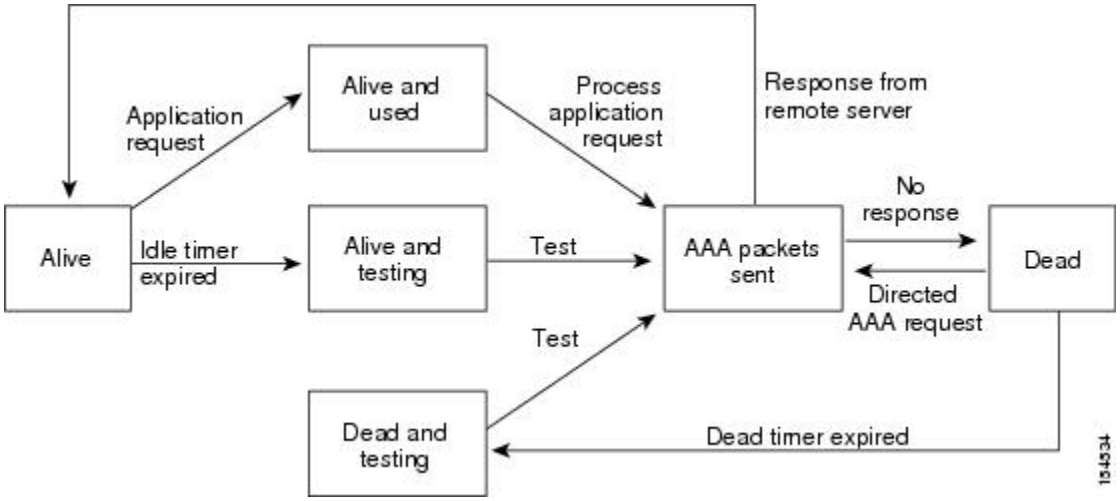
You can override the global preshared key assignment by explicitly using the key option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

Unresponsive TACACS+ servers are marked as dead and are not sent AAA requests. Dead TACACS+ servers are periodically monitored and brought back alive once they respond. This process confirms that a TACACS+ server is in a working state before real AAA requests are sent its way. The following figure shows how a

TACACS+ server state change generates a Simple Network Management Protocol (SNMP) trap and an error message showing the failure before it impacts performance.

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.

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 TACACS+ 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

The Cisco TACACS+ 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 = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use TACACS+ servers for authentication, the TACACS+ protocol directs the TACACS+ 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:

- Shell—Protocol used in access-accept packets to provide user profile information.
- Accounting—Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The following attributes are other supported:

- roles—Lists all the roles to which the user belongs. The value consists of a string that lists the role names delimited by white space. This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value.
- accountinginfo—Stores accounting information in addition to the attributes covered by a standard TACACS+ accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the TACACS+ client on the switch. It can be used only with the accounting protocol data units (PDUs).

Prerequisites for TACACS+

- Obtain the IP addresses or hostnames for the TACACS+ servers.
- Obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus 1000V is configured as a TACACS+ client of the AAA servers.
- You have already configured AAA, including remote TACACS+ authentication.

Guidelines and Limitations for TACACS+

- You can configure a maximum of 64 TACACS+ servers.
- The logging level for TACACS + must be set to 5.

Default Settings for TACACS+

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

Configuring TACACS+

The following flowchart guides you through the TACACS+ configuration process.

Note Be aware that the Cisco Nexus 1000V commands might differ from the Cisco IOS commands.

Figure 4: Configuring TACACS+ Flowchart

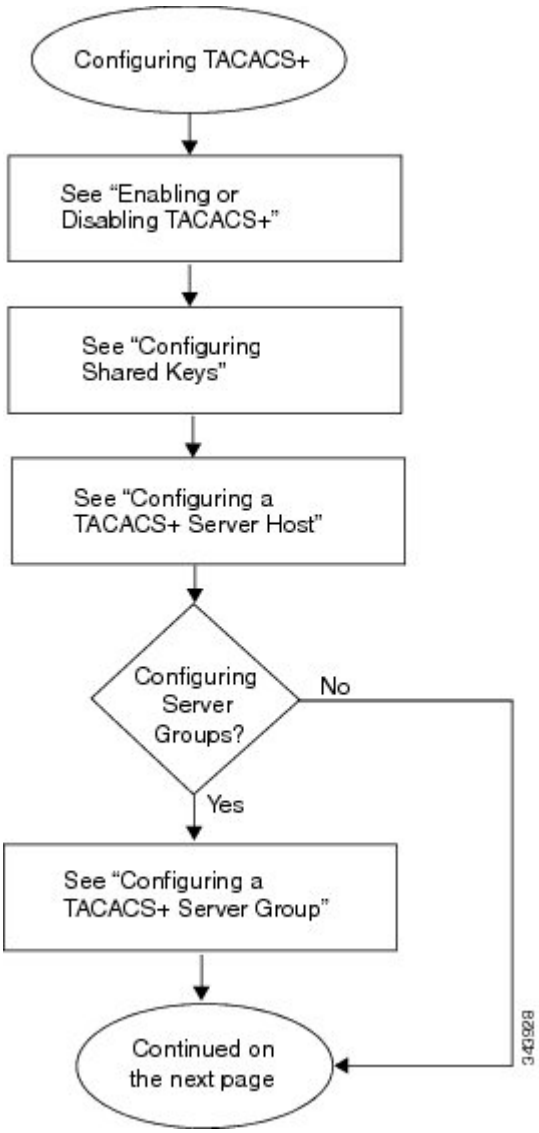


Figure 5: Configuring TACACS+ Flowchart (continued)

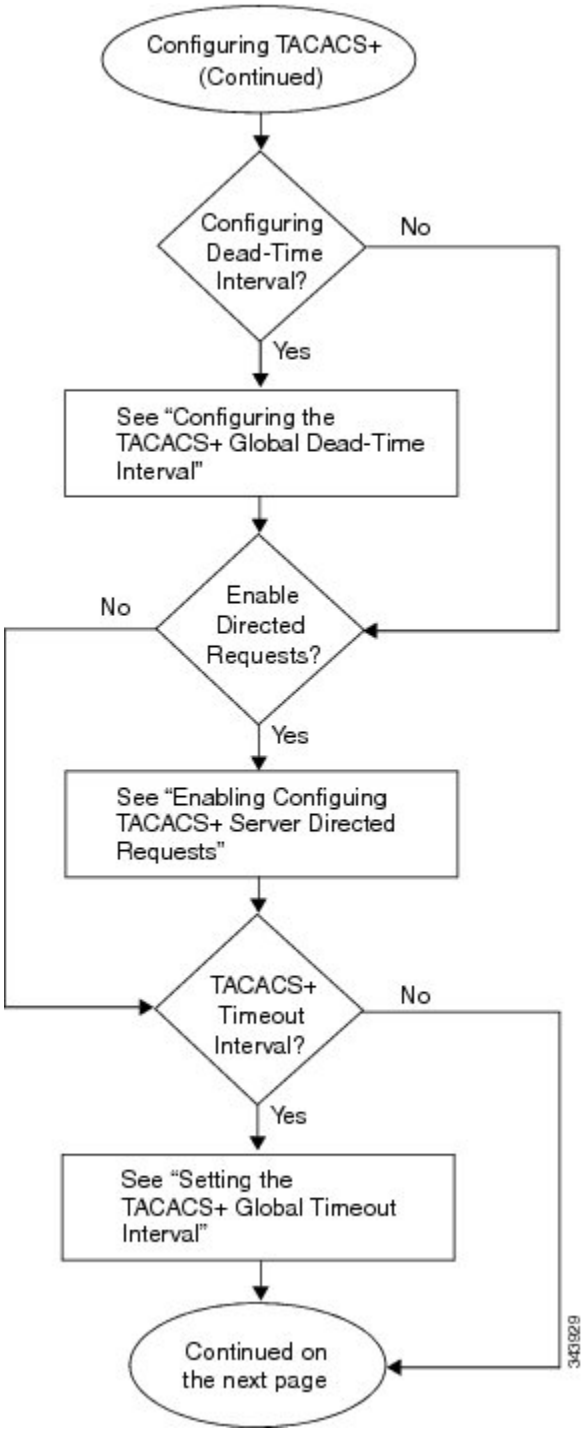
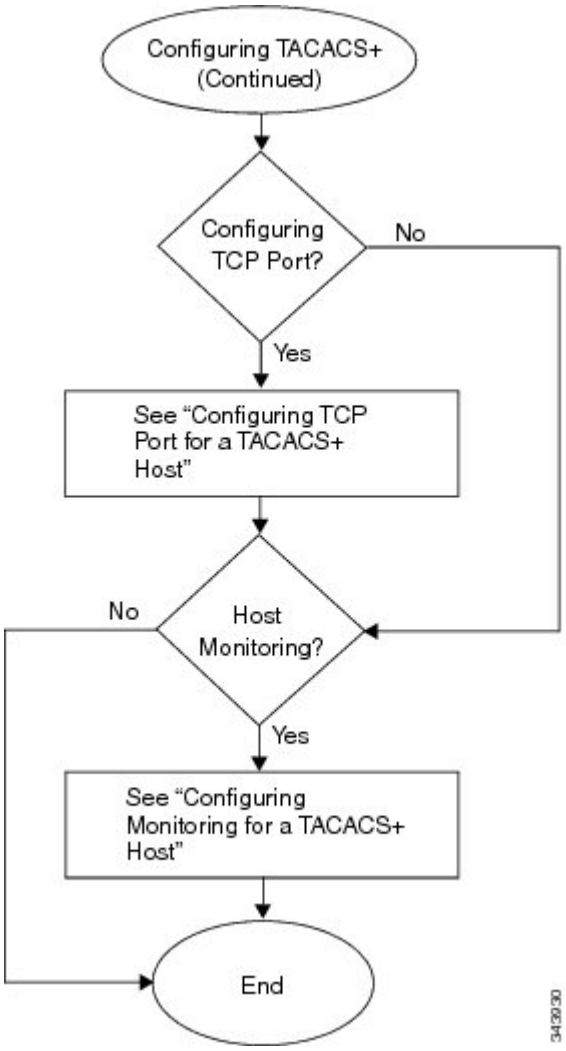


Figure 6: Configuring TACACS+ Flowchart (continued)



Enabling or Disabling TACACS+

By default, TACACS+ is disabled. You must explicitly enable the TACACS+ feature to access the configuration and verification commands that support TACACS+ authentication.

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

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **tacacs+ enable**
3. switch(config)# **exit**
4. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **tacacs+ enable**	Enables or disables TACACS+.
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

The following example enables TACACS+:

```
switch# configure terminal
switch(config)# tacacs+ enable
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Shared Keys

By default, no global key is configured.

You can configure the following:

- The global key, or a secret text string shared between the Cisco Nexus 1000V and all TACACS+ server hosts
- The key, or secret text string shared between the Cisco Nexus 1000V and a single TACACS+ server host

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Know the key for the TACACS+ server host(s).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server key** [**0** | **7**] *global_key*
3. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *shared_key*
4. switch(config)# **exit**
5. (Optional) switch(config)# **show tacacs-server**
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode. Do one of the following: • To configure a global key for all TACACS+ server hosts, continue to the next step. • To configure a key for a single TACACS+ server host, go to Step 3.
Step 2	switch(config)# **tacacs-server key** [**0** | **7**] *global_key*	Designates the global key shared between the Cisco Nexus 1000Vand the TACACS+ server hosts. 0—Specifies that a clear text string (key) follows. 0 is the default. 7—Specifies that an encrypted string (key) follows. *global_key*—The global key. The key is a string of up to 63 characters. By default, no global key is configured. Go to Step 4.
Step 3	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *shared_key*	Designates the key shared between the Cisco Nexus 1000V and this specific TACACS+ server host. 0—Specifies that a clear text string (key) follows. 0 is the default. 7—Specifies that an encrypted string (key) follows. *global_key*—The global key. The key is a string of up to 63 characters. This shared key is used instead of the global shared key.
Step 4	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 5	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration. **Note** The global shared key is saved in encrypted form in the running configuration. To display the key, use the show running-config command.

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

The following example configures shared keys:

```
switch# configure terminal
switch(config)# tacacs-server key 0 QsEFtkI#
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:5
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
switch# copy running-config startup-config
```

Configuring a TACACS+ Server Host

All TACACS+ server hosts are added to the default TACACS+ server group.

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the shared key.
- Know the IP addresses or the hostnames for the remote TACACS+ server hosts.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}
3. switch(config)# **exit**
4. (Optional) switch(config)# **show tacacs-server**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}	Configures the server IP address or hostname as a TACACS+ server host.

	Command or Action	Purpose
		ipv4-address—The IP address for the TACACS+ server. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

The following example configures a TACACS+ server host:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2
switch(config)# exit
switch# show tacacs-server
timeout value:5
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
switch# copy running-config startup-config
```

Configuring a TACACS+ Server Group

You can configure a TACACS+ server group whose member servers share authentication functions.

After you configure the TACACS+ server group, the server members are tried in the same order in which you configured them.

A TACACS+ server group can provide a failover if one server fails to respond. If the first server in the group fails, the next server in the group is tried until a server responds. Multiple server groups can provide failovers for each other in this same way.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that all servers added to a TACACS+ server group use the TACACS+ protocol.
- Configure the preshared keys.
- Enable TACACS+ for authentication.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa group server tacacs+** *group-name*
3. switch(config-tacacs+)# **server** {*ipv4-address* | *hostname*}
4. (Optional) switch(config-tacacs+)# **deadtime** *minutes*
5. (Optional) switch(config-tacacs+)# **use-vrf** *vrf-name*
6. (Optional) switch(config-tacacs+)# **source-interface** {*interface-type*} {*interface-number*}
7. (Optional) switch(config-tacacs+)# **show tacacs-server groups**
8. (Optional) switch(config-tacacs+)# **copy running-config startup-config**

DETAILED STEPS

	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 with the specified name and places you into the TACACS+ configuration mode for that group. *group-name*—The name of the TACACS+ server group.
Step 3	switch(config-tacacs+)# **server** {*ipv4-address* | *hostname*}	Configures the TACACS+ server hostname or IP address as a member of the TACACS+ server group. *ipv4-address*—The IP address for the TACACS+ server. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. **Note** If the specified TACACS+ server is not found, configure it using the tacacs-server host command and retry this command.
Step 4	switch(config-tacacs+)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time for this TACACS+ group. *minutes*—The dead time, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes. **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 5	switch(config-tacacs+)# **use-vrf** *vrf-name*	(Optional) Specifies the virtual routing and forwarding (VRF) instance to use to contact this server group. *vrf-name*—The name of the VRF instance.
Step 6	switch(config-tacacs+)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) Specifies a source interface to be used to reach the TACACS+ server. • loopback—Virtual interface number. The range is from 0 to 1023. • mgmt—Management interface 0.

	Command or Action	Purpose
		• null—Null interface 0. • port-channel—Port channel number. The range is from 1 to 4096.
Step 7	switch(config-tacacs+)# **show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration.
Step 8	switch(config-tacacs+)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures 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+)# use-vrf management
switch(config-tacacs+)# source-interface mgmt0
switch(config-tacacs+)# show tacacs-server groups
total number of groups:1

following TACACS+ server groups are configured:
        group TacServer:
                server 10.10.2.2 on port 49
                deadtime is 30
                vrf is management
switch# copy running-config startup-config
```

Enabling TACACS+ Server Directed Requests

You can designate the TACACS+ server to receive authentication requests. This is called a directed-request.

When directed requests are enabled, the user can log in as username@vrfname:hostname, where *vrfname* is the VRF to use and *hostname* is the name of a configured TACACS+ server.

Note User-specified logins are supported only for Telnet sessions.

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server directed-request**
3. switch(config)# **exit**
4. (Optional) switch(config)# **show tacacs-server directed-request**
5. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Enables use of directed requests for specifying the TACACS+ server to send an authentication request to when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server directed-request**	(Optional) Displays the TACACS+ directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

The following example enables TACACS+ server directed requests:

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

Setting the TACACS+ Global Timeout Interval

You can set the interval in seconds that the Cisco Nexus 1000V waits for a response from any TACACS+ server before declaring a timeout.

The timeout specified for an individual TACACS+ server overrides the global timeout interval.

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server timeout** *seconds*
3. switch(confi)# **exit**
4. (Optional) switch(config)# **show tacacs-server**
5. (Optional) switch(confi)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server timeout** *seconds*	Specifies the interval in seconds that the Cisco Nexus 1000V waits for a response from a server. *seconds*—The timeout interval, in seconds. The range is from 1 to 60 seconds. The default value is 5 seconds.
Step 3	switch(confi)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(confi)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example sets the TACACS+ global timeout interval:

```
switch# configure terminal
switch(config)# tacacs-server timeout 10
switch(config)# exit
switch# n1000v# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
switch# copy running-config startup-config
```

Setting a Timeout Interval for an Individual TACACS+ Host

You can set the interval in seconds that the Cisco Nexus 1000V waits for a response from a specific TACACS+ server before declaring a timeout. This setting is configured per TACACS+ host.

The timeout setting for an individual TACACS+ server overrides the global timeout interval.

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *hostname*} **timeout** *seconds*
3. switch(config)# **exit**
4. (Optional) switch(config)# **show tacacs-server**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *hostname*} **timeout** *seconds*	Specifies the timeout interval for a specific server. *ipv4-address*—The IP address for the TACACS+ server. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *seconds*—The timeout interval, in seconds. The range is from 1 to 60 seconds. The default value is the global timeout interval.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example sets a timeout interval for an individual TACACS+ host:

```
switch# config terminal
switch(config)#  tacacs-server host 10.10.2.2 timeout 10
switch(config)# exit
switch# n1000v# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
```

```
                timeout:10
switch# copy running-config startup-config
```

Configuring the TCP Port for a TACACS+ Host

You can configure a TCP port other than port 49 (the default for TACACS+ requests).

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **port** *tcp-port*
3. switch(config)# **exit**
4. (Optional) switch(config)# **show tacacs-server**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **port** *tcp-port*	Specifies the TCP port to use. *ipv4-address*—The IP address for the TACACS+ server. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *tcp-port*—The TCP port. The range is from 1 to 65535. The default value is 49.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures the TCP port for a TACACS+ host:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2 port 2
switch(config)# exit
```

```
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Configuring Monitoring for a TACACS+ Host

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.
- Know that the idle timer specifies how long a TACACS+ server should remain idle (receiving no requests) before sending it a test packet.
- Know that the default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not done.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]] }
3. switch(config)# **tacacs-server dead-time** *minutes*
4. switch(config)# **exit**
5. (Optional) switch(config)# **show tacacs-server**
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]] }	Configures server monitoring. *ipv4-address*—The IP address for the TACACS+ server. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *minutes*—The idle time interval, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes. For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.

	Command or Action	Purpose
		password—The user's password. The default value is test. *name*—The username to use when connecting to the TACACS+ server. The default value is test. To protect network security, we recommend that you assign a username that is not already in the TACACS+ database.
Step 3	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the duration of time in minutes before checking a TACACS+ server that was previously unresponsive. *minutes*—The dead time interval, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes.
Step 4	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 5	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures monitoring for a TACACS+ host:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2 test username pvk2 password a3z9yjqz7 idle-time
 3
switch(config)# tacacs-server dead-time 5
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:5
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Configuring the TACACS+ Global Dead-Time Interval

You can configure the interval to wait before sending a test packet to a previously unresponsive server.

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

Before You Begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server deadtime** *minutes*
3. switch(config)# **exit**
4. (Optional) switch(config)# **show tacacs-server**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	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. *minutes*—The dead-time interval, in minutes. The range is from 0 to 1440 minutes. The default value is 0 minutes.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures the TACACS+ global dead-time interval:

```
switch# configure terminal
switch(config)#  tacacs-server deadtime 5
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:5
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Displaying Statistics for a TACACS+ Host

Use the following command to display statistics for a TACACS+ host:

Command	Description
show tacacs-server statistics {*hostname* | *ipv4-address*}	Displays the statistics for a TACACS+ host. *hostname*—The hostname for the TACACS+ server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *ipv4-address*—The IP address for the TACACS+ server.

Configuration Example for TACACS+

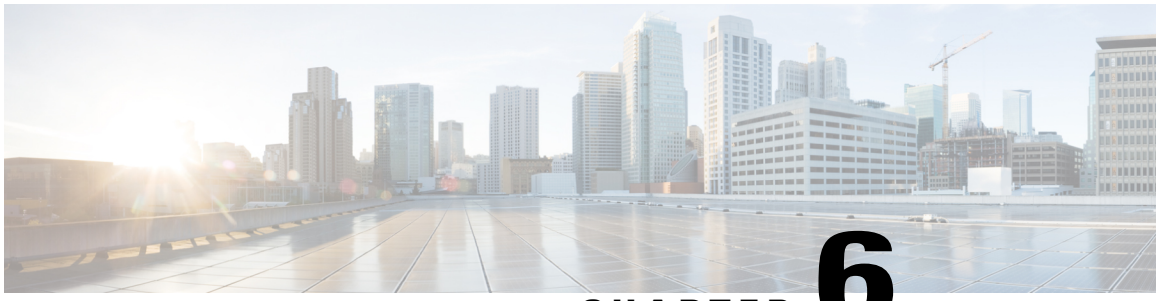
The following example configures a TACACS+ server:

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

Feature History for TACACS+

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
TACACS+	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 6

Configuring SSH

This chapter contains the following sections:

Information About SSH

SSH Server

You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection. SSH uses strong encryption for authentication. The SSH server can operate with publicly and commercially available SSH clients.

TACACS+ user authentication and locally stored usernames and passwords are supported for SSH.

SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a secure, encrypted connection to any device that runs the SSH server. This connection provides an encrypted outbound connection. With authentication and encryption, the SSH client produces secure communication over an insecure network.

The SSH client works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communication. 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)

Be sure to have an SSH server key-pair with the correct version before enabling the SSH service. Generate the SSH server key-pair according to the SSH client version used. The SSH service accepts two 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.

By default, an RSA key that uses 1024 bits is generated.

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.

Prerequisites for SSH

- Configure IP on a Layer 3 interface, out-of-band on the mgmt 0 interface.
- Before enabling the SSH server, obtain the SSH key.

Guidelines and Limitations for SSH

- Only SSH version 2 (SSHv2) is supported.
- SSH is enabled by default.
- Cisco NX-OS commands might differ from the Cisco IOS commands.

Default Settings

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 2048 bits
RSA key bits for generation	1024

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.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature ssh**
3. switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}
4. switch(config)# **feature ssh**
5. (Optional) switch# **show ssh key**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables SSH.
Step 3	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key. **dsa—**If specified, the ssh command uses the Digital Signature Algorithm (DSA) public key algorithm. **rsa—**If specified, the ssh command uses the Ron Rivest, Adi Shamir, and Leonard Adleman (RSA) public key algorithm.

	Command or Action	Purpose
		bits—The number of bits used to generate the key. The range is from 768 to 2048. The default value is 1024. **force**—Replaces an existing key.
Step 4	switch(config)# **feature ssh**	Enables SSH.
Step 5	switch# **show ssh key**	(Optional) Displays the SSH server keys.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example generates SSH server keys:

```
switch# configure terminal
switch(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
switch(config)# ssh key dsa force
generating dsa key(1024 bits).....
.
generated dsa key
n1000v(config)# feature ssh
n1000v(config)# show ssh key
*************************************
rsa Keys generated:Sun Jul 27 15:18:46 2008

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXK
fVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6Uwa
GID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+f
FzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4
GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==

bitcount:2048
fingerprint:
fd:ca:48:73:b9:ee:e7:86:9e:1e:40:46:f1:50:1d:44
*************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
*************************************
```

Configuring a User Account with a Public Key

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

- OpenSSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Configuring an OpenSSH Key

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

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

- OpenSSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Before You Begin

- Log in to the CLI in EXEC mode.
- Generate an SSH public key in OpenSSH format.
- Have an existing user account.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **username** *username* **sshkey** *ssh-key*
3. switch(config)# **exit**
4. (Optional) switch# **show user-account**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enter global configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in OpenSSH format with an exiting user account. *username*—The username of the existing user account. *ssh-key*—The SSH public key to use. To create a user account use the **username** *name* **password** *pwd* command. *name*—The name to assign to the user account. *username*—The password to assign to the user account.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 4	switch# **show user-account**	(Optional) Displays the user account configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures an OpenSSH key.

```
switch# configure terminal
switch(config)# username user1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyK
cb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5aw
fVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8
QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuD
YSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby
/ZWx/clTYiXQR1X1VfhQ==
switch(config)# exit
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdH
Ha/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6
/n3FVroyRwupMki6mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EI
r/0XIP1mqTsrqTsmjZ2vLk+fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m
9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1Vf
hQ==
switch# copy running-config startup-config
```

Configuring IETF or PEM Keys

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

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

- OpenSSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Before You Begin

- Log in to the CLI in EXEC mode
- Generate an SSH public key in one of the following formats:
  - IETF SECSH format
  - Public Key Certificate in PEM format

SUMMARY STEPS

1. switch# **copy** *server-file* **bootflash:***filename*
2. switch# **configure terminal**
3. switch(config)# **username** *username* **sshkey file bootflash:***filename*
4. switch(config)# **exit**
5. (Optional) switch# **show user-account**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:***filename*	Downloads the file containing the SSH key from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP. *filename*—The name of the file that contains the SSH key.
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **username** *username* **sshkey file bootflash:***filename*	Configures the SSH public key. *username*—The username of the existing user account. *filename*—The name of the file that contains the SSH key.
Step 4	switch(config)# **exit**	Exits global configuration mode and returns you to EXEC mode.
Step 5	switch# **show user-account**	(Optional) Displays the user account configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example configures an SSH key to use when logging in with the specified user account:

```
switch# copy tftp://10.78.1.10/secsh_file.pub bootflash:secsh_file.pub vrf management
Trying to connect to tftp server......
Connection to server Established.
|
TFTP get operation was successful
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:user2
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/
ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6
mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+
fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJN
```

```
U1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/c1TYiXQR1X1VfhQ==
switch# copy running-config startup-config
```

Starting SSH Sessions

You can start SSH sessions using IP to connect to remote devices.

Before You Begin

- Log in to the CLI in EXEC mode.
- Obtain the hostname and, if needed, the username, for the remote device.
- Enable the SSH server on the remote device.

SUMMARY STEPS

1. switch# **ssh** [**root@**] {*ip-address* | *hostname* } [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **ssh** [**root@**] {*ip-address* | *hostname* } [**vrf** *vrf-name*]	Creates an SSH IP session to a remote device using the device's IP address. *ip-address*—The IP address of the remote device. *hostname*—The hostname of the remote device. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *vrf-name*—The name of the virtual routing and forwarding (VRF) instance. The default value is the default VRF instance.

The following example starts an SSH session:

```
switch# ssh root@172.28.30.77
root@172.28.30.77's password:
Last login: Sat Dec 4 11:07:23 2013 from 171.70.209.64
```

Clearing SSH Hosts

You can clear from your account the list of trusted SSH servers that were added when you downloaded a file from a server using SCP or SFTP, or when you started an SSH session to a remote host.

SUMMARY STEPS

1. switch# **clear ssh hosts**

DETAILED STEPS

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

Disabling the SSH Server

You can disable the SSH server to prevent SSH access to the switch. By default, the SSH server is enabled.

If you disable SSH, you must first generate an SSH server key before you can enable it again.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature ssh**
3. (Optional) switch(config)# **show ssh server**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

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

The following example disables the SSH server:

```
switch# configure terminal
switch(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
switch(config)#  show ssh server
ssh is not enabled
switch(config)# copy running-config startup-config
```

Deleting SSH Server Keys

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

If you disable SSH, you must first generate an SSH server key before you can enable it again.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature ssh**
3. switch(config)# **no ssh key** [**dsa** | **rsa**]
4. (Optional) switch(config)# **show ssh key**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	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. **dsa—**If specified, the ssh command deletes the Digital Signature Algorithm (DSA) public key. **rsa—**If specified, the ssh command deletes the Ron Rivest, Adi Shamir, and Leonard Adleman (RSA) public key. The default is to delete all of the SSH keys.
Step 4	switch(config)# **show ssh key**	(Optional) Displays the SSH server key configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example deletes the SSH server keys:

```
switch# configure terminal
switch(config)# no feature ssh
switch(config)# no ssh key rsa
switch(config)# show ssh key
*************************************
rsa Keys generated:Sun Jul 27 15:18:46 2008

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXK
fVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6Uwa
GID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+f
```

```
FzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4
GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==

bitcount:2048
fingerprint:
fd:ca:48:73:b9:ee:e7:86:9e:1e:40:46:f1:50:1d:44
*************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
*************************************
mcs-srvr43(config)# no ssh key rsa
mcs-srvr43(config)# show ssh key
*************************************
could not retrieve rsa key information
*************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
*************************************
mcs-srvr43(config)# no ssh key dsa
mcs-srvr43(config)# show ssh key
*************************************
could not retrieve rsa key information
*************************************
could not retrieve dsa key information
*************************************
no ssh keys present. you will have to generate them
*************************************
```

Clearing SSH Sessions

You can clear SSH sessions from the device.

Before You Begin

Log n to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*
3. (Optional) switch# **show users**

DETAILED STEPS

	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. *vty-line*—The Virtual Teminal Line (VTY) to clear.
Step 3	switch# **show users**	(Optional) Displays user session information.

The following example clears the SSH sessions from the device:

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/0        Jul 28 09:49 00:02       28556 (10.21.148.122)
admin    pts/1        Jul 28 09:46   .         28437 (::ffff:10.21.148.122)*
switch# clear line 0
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/1        Jul 28 09:46   .         28437 (::ffff:10.21.148.122)*
mcs-srvr43(config)#
```

Verifying the SSH Configuration

You can verify the configuration with the following commands:

Command	Purpose
show ssh key [**dsa** | **rsa**]	Displays SSH server key-pair information. **dsa—**If specified, displays the Digital Signature Algorithm (DSA) public key information. **rsa—**If specified, displays the Ron Rivest, Adi Shamir, and Leonard Adleman (RSA) public key information. The default is to display information for all of the SSH keys.
show running-config security [**all**]	Displays the SSH and user account configuration in the running configuration. **all—**Displays the default values for the SSH and user accounts.
show ssh server	Displays the SSH server configuration.

Configuration Example for SSH

The following example shows how to configure SSH with an OpenSSH key:

1. Disable the SSH server:

```
switch# configure terminal
switch(config)# no feature ssh
```

2. Generate an SSH server key:

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

3. Enable the SSH server:

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

4. Display the SSH server key:

```
switch(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2007

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr+MZm99n2U0
ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39HmXL6VgpRVn1XQFiBwn4
na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

bitcount:1024
fingerprint:
51:6d:de:1c:c3:29:50:88:df:cc:95:f0:15:5d:9a:df
*************************************
could not retrieve dsa key information
*************************************
```

5. Specify the SSH public key in OpenSSH format:

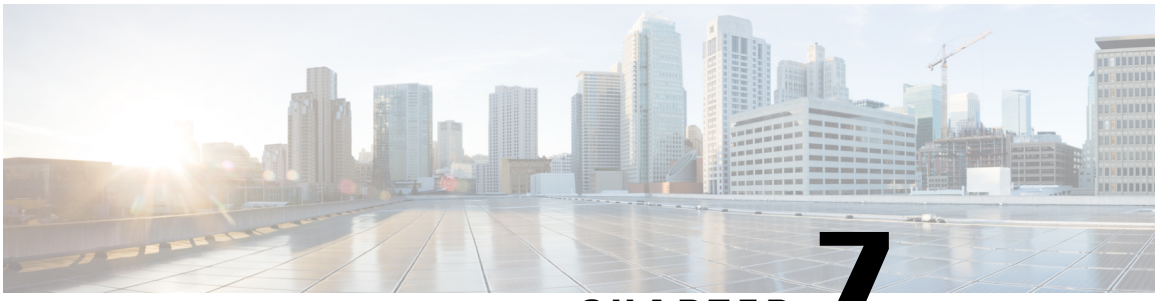
```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=
```

6. Save the configuration:

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

Feature History for SSH

Feature Name	Releases	Feature Information
SSH	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 7

Configuring Telnet

This chapter contains the following sections:

Information About the Telnet Server

The Telnet protocol enables you to set up 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 pass the keystrokes from one device to the other. Telnet can accept either an IPv4 address or a domain name as the remote device address.

Prerequisites for Telnet

You have configured IP on a Layer 3 interface, out of band on the mgmt 0 interface.

Guidelines and Limitations for Telnet

- The Telnet server is disabled by default.
- Cisco NX-OS commands may differ from Cisco IOS commands.

Default Setting for Telnet

Parameter	Default
Telnet server	Disabled

Configuring Telnet

Enabling the Telnet Server

The Telnet server is enabled by default, but you can reenable the server if necessary.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature telnet**
3. (Optional) switch(config)# **show telnet server**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature telnet**	Enables the Telnet server.
Step 3	switch(config)# **show telnet server**	(Optional) Displays the Telnet server configuration.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example enables the Telnet server:

```
switch# configure terminal
switch(config)# feature telnet
switch(config)# show telnet server
telnet service enabled
switch(config)# copy running-config startup-config
```

Starting an IP Telnet Session to a Remote Device

Before You Begin

- Log in to the CLI in EXEC mode.
- Verify that the Telnet server is enabled and that the server is also enabled on the remote device.
- Obtain the hostname for the remote device and, if needed, the username on the remote device.

SUMMARY STEPS

1. switch# **telnet** {*ip address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **telnet** {*ip address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]	Creates an IP Telnet session to the specified destination. The keywords and arguments are as follows: • *port-number*—Port number to use for this session. The range is from 1 to 65535. The default port number is 23. • *vrf-name*—Default virtual routing and forwarding (VRF) instance.

Clearing Telnet Sessions

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*
3. (Optional) switch# **show users**

DETAILED STEPS

	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.

	Command or Action	Purpose
		vty-line—The Virtual Teminal Line (VTY) to clear.
Step 3	switch# **show users**	(Optional) Displays user session information.

The following example clears a Telnet session:

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/1        Jul 28 14:04   .         31453 (::ffff:171.70.209.8)
admin    pts/2        Jul 28 14:04   .         31475 (171.70.209.8)*
switch# clear line 1
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/2        Jul 28 14:04   .         31475 (171.70.209.8)*
switch#
```

Verifying the Telnet Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
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.
show telnet server	Displays the Telnet server configuration.
show hosts	Displays the configuration details for current hosts.

Example 1: show running-config security all

The following example displays the user account configuration in the running configuration for :

```
switch(config)# show running-config security all

!Command: show running-config security all
!Time: Tue Sep 17 22:04:20 2013

version 5.2(1)SK1(2.1)
feature telnet
feature http-server
no feature scp-server
no feature sftp-server
feature ssh

username adminbackup password 5 ! role network-operator
username admin password 5 $1$qnBf/DZs$SADurdd7yy/VMA19E./N11 role network-admin
username admin keypair rsa
username ajidas password 5 $1$L8yMjMMy$xIGNCA.CkgnVY70nBqnhr1 expire 2020-12-30
role network-operator
```

```
password strength-check

banner motd #Nexus 1000v Switch
#

ssh key rsa 2048
no ssh key dsa
```

Example 2: show telnet server

The following example displays the Telnet server configuration:

```
switch(config)# show telnet server
telnet service enabled
```

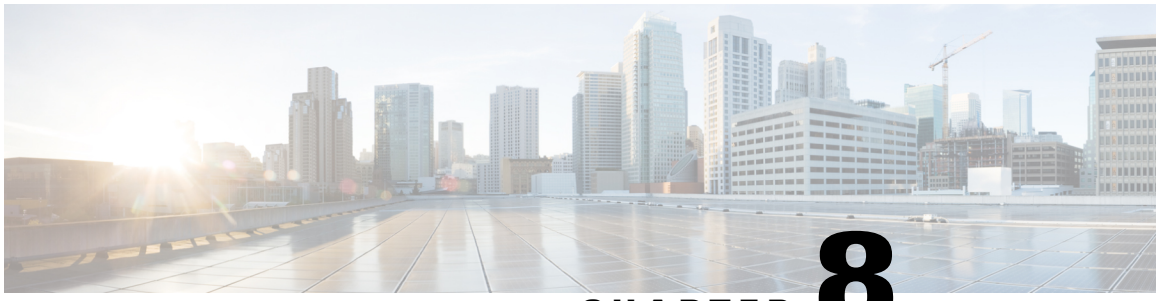
Example 3: show hosts

The following example displays the configuration details for current hosts:

```
switch(config)# show hosts
DNS lookup enabled
Name/address lookup uses domain service
Name servers are 255.255.255.255


Host                    Address
host1                   192.0.2.0
host2                   198.51.100.0
host3                   203.0.113.0
```

Feature History for Telnet

Feature Name		Feature Information
Telnet	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 8

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

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 device determines that an ACL applies to a packet, the device tests the packet against the conditions of all rules. The rule determines whether the packet is to be permitted or denied. If there is no match to any of the specified rules, then the device denies the packet. The device 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 can use ACLs to disallow HTTP traffic from a high-security network to the Internet. You can also use ACLs to allow HTTP traffic to a specific site using the IP address of the site to identify it in an IP ACL.

ACL Types and Applications

An ACL is considered a port ACL when you apply it to one of the following:

- Ethernet interface
- vEthernet interface

When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on that trunk port.

Order of ACL Application

When the device processes a packet, it determines the forwarding path of the packet. The device applies the ACLs in the following order:

1. Ingress port ACL
2. Egress port ACL

Rules

Rules are what you create, modify, and remove when you configure how an access control list (ACL) filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to all VEMs.

You can create rules in ACLs in access-list configuration mode by using the **permit** or **deny** command. The device 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 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

ACLs allow you to identify traffic by protocol. You can specify some protocols by name. For example, in an IP ACL, you can specify ICMP by name.

In IP ACLs, you can specify protocols by the integer that represents the Internet protocol number. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

Implicit Rules

ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules. Implicit rules ensure that unmatched traffic is denied, regardless of the protocol specified in the Layer 2 header of the traffic.

All IPv4 ACLs include the following implicit rule that denies unmatched IP traffic:

```
deny ip any any
```

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IP 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

Sequence Numbers

The 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 by 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, you can 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.

Statistics

The device can maintain global statistics for each rule that you configure. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

Note The device does not support interface-level ACL statistics.

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

Prerequisites for IP ACLs

- You must be familiar with IP addressing and protocols to configure IP ACLs.
- You must be familiar with the port profile interface types that you want to configure with ACLs.

Guidelines and Limitations for IP ACLs

ACLs are not supported in port channels.

Default Settings for IP ACLs

Parameters	Default
IP ACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs.

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 ACL on the device and add rules to it.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **ip access-list** *name*
3. switch(config-acl)# [*sequence-number*] { **permit** | **deny**}*protocol source destination*
4. (Optional) switch(config-acl)# **statistics per-entry**
5. (Optional) switch(config-acl)# **show ip access-lists** *name*
6. (Optional) switch(config-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ip access-list** *name*	Creates the named IP ACL (up to 64 characters in length) and enters IP ACL configuration mode. The **no** option removes the specified access list.
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 from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V Command Reference* for more information.
Step 4	switch(config-acl)# **statistics per-entry**	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	switch(config-acl)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 6	switch(config-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example creates an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    10 permit ip 192.168.2.0/24 any
switch(config-acl)# copy running-config startup-config
```

Changing an IP ACL

You can add and remove rules in an existing IPv4 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and create it again 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.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip access-list** *name*
3. (Optional) switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}
5. (Optional) switch(config-acl)# [**no**] **statistics per-entry**
6. (Optional) switch(config-acl)# **show ip access-lists** *name*
7. (Optional) switch(config-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the specified ACL.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	(Optional) 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 from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for KVM Reference Guide* for more information.
Step 4	switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for KVM Reference Guide* for more information.
Step 5	switch(config-acl)# [**no**] **statistics per-entry**	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.

	Command or Action	Purpose
		The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	switch(config-acl)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 7	switch(config-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example changes an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-acl)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    10 permit ip 192.168.2.0/24 any
switch(config-acl)# ip access-list acl-01
switch(config-acl)# no 10
switch(config-acl)# no statistics per-entry
switch(config-acl)# show ip access-lists acl-01

IPV4 ACL acl-01
switch(config-acl)# copy running-config startup-config
```

Removing an IP ACL

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, that is, an empty ACL with an implicit rule of "deny ip any any." Use the **show ip access-lists** command with the **summary** keyword to find the interfaces on which the IP ACL is configured.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know whether the ACL is applied to an interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no ip access-list** *name*
3. (Optional) switch(config)# **show ip access-list** *name* **summary**
4. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration. name—Specifies the name of the ACL.
Step 3	switch(config)# **show ip access-list** *name* **summary**	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces. name—Specifies the name of the ACL.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

The following example removes an IP ACL:

```
switch# configure terminal
switch(config)# no ip access-list acl-01
switch(config)# show ip access-lists acl-01 summary
switch(config)# copy running-config startup-config
```

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# **show ip access-lists** *name*
3. switch(config)# **resequence ip access-list** *name starting-sequence-number increment*
4. switch(config)# **show ip access-lists** *name*
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration. *name*—Specifies the name of the ACL.
Step 3	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL. *name*—Specifies the ACL name. The maximum length is 64 characters. *starting-sequence-number*—Specifies the sequence number of the first rule. The range is from 1 to 4294967295. *increment*—Specifies the amount by which to increment a sequence number to get the sequence number of each subsequent rule. The range is from 1 to 4294967295.
Step 4	switch(config)# **show ip access-lists** *name*	Displays the IP ACL configuration. *name*—Specifies the name of the ACL.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example changes the sequence numbers in an IP ACL:

```
switch# configure terminal
Enter configuration commands one command per line. End with CNTL/Z.
switch(config)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    10 permit ip 192.168.2.0/24 any
    20 permit ip 192.168.5.0/24 any
switch(config)# resequence ip access-list acl- 01 100 10
switch(config)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    100 permit ip 192.168.2.0/24 any
    110 permit ip 192.168.5.0/24 any
switch# copy running-config startup-config
```

Applying an IP ACL as a Port ACL

You can apply an IPv4 ACL to a physical Ethernet interface or a virtual Ethernet interface. ACLs that are applied to these interface types are considered port ACLs. An IP ACL can also be applied on a port-profile attached to a physical Ethernet interface or virtual Ethernet interface.

ACLs cannot be applied on a port-channel interface. However, they can be applied on a physical Ethernet interface that is not part of the port channel.

If ACL does not exist or have no rules when applied on the ports, then traffic is implicitly denied on these ports.

Before You Begin

- Log in to the CLI in EXEC mode.
- Apply one port ACL to an interface.
- Verify 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.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet** *slot-number* | **vethernet** *interface-number*}
3. switch(config-if)# **ip port access-group** *access-list* [**in** | **out**]
4. (Optional) switch(config-if)# **show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot-number* | **vethernet** *interface-number*}	Enters interface configuration mode for the specified interface. Port ACLs are not supported on a port-channel interface and physical Ethernet interface that is a member of the port channel. **ethernet**—Specifies the Ethernet IEEE 802.3z interface. *slot-number*—Specifies the slot number. The range is from 1 to 514. **vethernet**—Specifies the virtual Ethernet interface. *interface-number*—Specifies the interface number. The range is from 1 to 1048575.
Step 3	switch(config-if)# **ip port access-group** *access-list* [**in** | **out**]	Applies an inbound or outbound IPv4 ACL to the interface. You can apply one port ACL to an interface. *access-list*—Specifies the port ACL. The maximum length is 64 characters. **in**—Specifies inbound packets for the ACL. **out**—Specifies outbound packets for the ACL.
Step 4	switch(config-if)# **show running-config aclmgr**	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example applies an IP ACL as a port ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 1
switch(config-if)# ip port access-group acl-01 in
switch(config-if)# show running-config aclmgr

!Command: show running-config aclmgr
!Time: Wed Mar 13 02:19:05 2013

version 5.2(1)SK1(2.1)
ip access-list acl-01
    statistics per-entry
    100 permit ip 192.168.2.0/24 any
    110 permit ip 192.168.5.0/24 any

interface Vethernet1
    ip port access-group acl-01 in

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

Adding an IP ACL to a Port Profile

You can add an IP ACL to a port profile.

Before You Begin

- Log in to the CLI in EXEC mode.
- Create the IP ACL to add to this port profile and you know its name.
- If you are using an existing port profile, you have must have created it and you know its name.
- If you want to create a new port profile, you must know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- Know the name of the IP access control list that you want to configure for this port profile.
- Know the direction of the packet flow for the access list.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. switch(config-port-prof)# **ip port access-group** *access-list* {**in** | **out**}
4. (Optional) switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the specified port profile. **type ethernet—**Specifies the Ethernet type for the port profile. **type vethernet—**Specifies the vEthernet type for the port profile. *name***—**Specifies the port profile. The maximum length is 80 characters.
Step 3	switch(config-port-prof)# **ip port access-group** *access-list* {**in** | **out**}	Adds the named ACL to the port profile for either inbound or outbound traffic. *access-list***—**Specifies the port ACL. The maximum length is 64 characters. **in—**Specifies inbound packets for the ACL. **out—**Specifies outbound packets for the ACL.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification. **brief—**Specifies brief output. **expand-interface—**Applies the active port-profile configuration to an interface. **usage—**Lists the interfaces that are inherited by the port-profile. **name** *profile-name***—**Specifies the port profile name.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example adds an IP ACL to a port profile:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# port-profile vm_eth1
switch(config-port-prof)# ip port access-group acl-01 out
switch(config-port-prof)# end
switch# show port-profile name vm_eth1

port-profile vm_eth1
type: Vethernet
description:
status: enabled
max-ports: 32
min-ports: 1
inherit:
config attributes:
ip port access-group acl-01 out
no shutdown
evaluated config attributes:
ip port access-group acl-01 out
no shutdown
assigned interfaces:
port-group: vm_eth1
system vlans: none
capability l3control: no
capability iscsi-multipath: no
capability vxlan: no
capability l3-vn-service: no
```

```
port-profile role: none
port-binding: static

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

Applying an IP ACL to the Management Interface

You can apply an IPv4 ACL to the management interface, mgmt0.

Be sure 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.

If ACL does not exist or have no rules applied on the Virtual Supervisor Module (VSM) mgmt0, then all traffic is implicitly permitted.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface mgmt0**
3. switch(config-if)# [**no**] **ip access-group** *access-list* [**in** | **out**]
4. (Optional) switch(config-if)# **show ip access-lists** *access-list*
5. switch(config-if)# [**no**] **ip access-list match-local-traffic**
6. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface mgmt0**	Enters interface configuration mode for the management interface.
Step 3	switch(config-if)# [**no**] **ip access-group** *access-list* [**in** | **out**]	Applies a specified inbound or outbound IPv4 ACL to the interface. **no—**Removes the specified configuration. *access-list***—**Specifies the port ACL. The maximum length is 64 characters. **in—**Specifies inbound packets. for the ACL. **out—**Specifies outbound packets. for the ACL.
Step 4	switch(config-if)# **show ip access-lists** *access-list*	(Optional) Displays the ACL configuration. *access-list***—**Specifies the port ACL. The maximum length is 64 characters.
Step 5	switch(config-if)# [**no**] **ip access-list match-local-traffic**	Enables matching for locally generated traffic.

	Command or Action	Purpose
		This global command must be enabled for ACL rules to take effect when an ACL is applied in the egress direction on mgmt0 interface. **no—**Disables matching for locally generated traffic.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example applies an IP ACL to the management interface:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-acl)# interface mgmt 0
switch(config-if)# ip access-group acl-01 out
switch(config-if)# show ip access-lists acl-01 summary

IPV4 ACL acl-01
    Total ACEs Configured:1
    Configured on interfaces:
      mgmt0 - egress (Router ACL)
    Active on interfaces:
    mgmt0 - egress (Router ACL)
switch(config-if)# ip access-list match-local-traffic
switch(config)# copy running-config startup-config ACL)
```

Verifying the IP ACL Configuration

Use the following commands to verify the configuration:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including the IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists [*name*]	Displays all IPv4 access control lists or a named IPv4 ACL.
show ip access-list [*name*] **summary**	Displays a summary of all configured IPv4 ACLs or a named IPv4 ACL.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.

Monitoring IP ACLs

Use the following commands for IP ACL monitoring:

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration. If the IPv4 ACL configuration includes the **statistics per-entry** command, the **show ip access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL.

On a universal virtual Ethernet module (uVEM) host that has an ACL applied, use the following commands for IP ACL monitoring:

Command	Purpose
vemcmd show acl	Displays ACL IDs.
vemcmd show acl debug stats	Displays ACL debug statistics.
vemcmd show acl pinst	Displays the ACL policy instance.
vemcmd show acl pinst tables	Displays the ACL policy instance tables.

Configuration Example for IP ACL

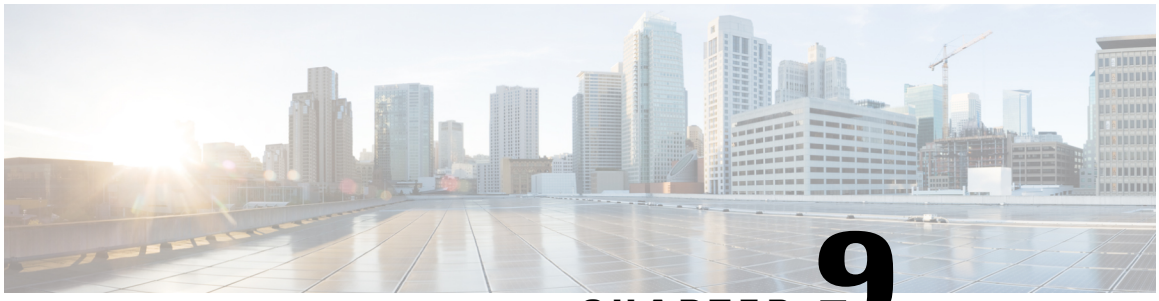
The following example shows how to create an IPv4 ACL named acl-01, apply the ACL as a port ACL on a physical Ethernet interface that is not a member of a port channel, and configure verification with match counters:

```
switch# configure terminal
Enter configuration commands one per line. End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# permit ip 192.168.5.0/24 any
switch(config-acl)# permit 22 any 10.105.225.225/27
switch(config-acl)# permit ip any 10.105.225.225/27
switch(config-acl)# statistics per-entry
switch(config-acl)# interface ethernet 3/5
switch(config-acl# ip port access-group acl-01 in
switch(config-acl)# show ip access-lists acl-01 summary

IPV4 ACL acl-01
  statistics per-entry
  Total ACEs Configured:4
  Configured on interfaces:
    Ethernet3/5 - ingress (Port ACL)
Active on interfaces:
    Ethernet3/5 - ingress (Port ACL)
switch(config-if)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    100 permit ip 192.168.2.0/24 any [match=0]
    110 permit ip 192.168.5.0/24 any [match=0]
    120 permit 22 any 10.105.225.225/27 [match=0]
    130 permit ip any 10.105.225.225/27 [match=44]
switch(config-if)# clear ip access-list counters acl-01
```

```
switch(config-if)# show ip access-lists acl-01

IPV4 ACL acl-01
    statistics per-entry
    100 permit ip 192.168.2.0/24 any [match=0]
    110 permit ip 192.168.5.0/24 any [match=0]
    120 permit 22 any 10.105.225.225/27 [match=0]
    130 permit ip any 10.105.225.225/27 [match=0]
switch(config-if)#
```

Feature History for IP ACLs

Feature History	Releases	Feature Information
IP ACLs	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 9

Configuring MAC ACLs

This chapter contains the following sections:

Prerequisites for MAC ACLs

- You must be familiar with MAC addressing and non-IP protocols to configure MAC ACLs.
- You must be familiar with the ACL concepts presented in this document.

Guidelines and Limitations for MAC ACLs

ACLs are not supported in port channels.

Default Settings for MAC ACLs

Parameters	Default
MAC ACLs	No MAC ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs.

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it. You can also use this procedure to add the ACL to a port profile.

Before You Begin

- Log in to the CLI in EXEC mode.
- Have a name to assign to the ACL that you are creating.
- Create a port profile if you want to add the ACL to it.

If you want to also add the ACL to a port profile, you must know the following:

- If you are using an existing port profile, you have already created it and you know its name.
- The interface type (Ethernet or vEthernet) and the name you want to give the port profile if you are creating a new port profile.
- The direction of packet flow for the access list.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **statistics per-entry**
5. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
6. (Optional) switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 3	switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for KVM Reference Guide* for more information.

	Command or Action	Purpose
Step 4	switch(config-mac-acl)# **statistics per-entry**	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) Displays the MAC ACL configuration for verification. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 6	switch(config-mac-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example creates a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
    statistics per-entry
    10 permit 00c0.4f00.0000 0000.00ff.ffff any
switch# copy running-config startup-config
```

Changing a MAC ACL

You can change an existing MAC ACL, such as to add or remove rules.

Use the **resequence** command to reassign sequence numbers, such as when adding rules between existing sequence numbers.

Before You Begin

- Log in to the CLI in EXEC mode.
- In an existing MAC ACL, you cannot change existing rules.
- In an existing MAC ACL, you can add and remove rules.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. (Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}
5. switch(config-mac-acl)# [**no**] **statistics per-entry**
6. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
7. switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 3	switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*	(Optional) Creates a rule in the MAC 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 **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for KVM Reference Guide* for more information. *sequence-number*—Specifies the sequence number. The range is from 1 to 4294967295.
Step 4	switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}	(Optional) Removes the rule that you specify from the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for KVM Reference Guide* for more information. *sequence-number*—Specifies the sequence number. The range is from 1 to 4294967295.
Step 5	switch(config-mac-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) Displays the MAC ACL configuration for verification. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 7	switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

The following example changes a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# no 10
switch(config-mac-acl)# no statistics per-entry
switch(config-mac-acl)# end
switch# show mac access-lists

MAC ACL acl-mac-01
        20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch# copy running-config startup-config
```

Removing a MAC ACL

You can remove a MAC 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.

To find the interfaces that a MAC ACL is configured on, use the **show mac access-lists** command with the **summary** keyword.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know whether the ACL is applied to an interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no mac access-list** *name*
3. (Optional) switch(config)# **show mac access-lists** *name* **summary**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no mac access-list** *name*	Removes the specified MAC ACL from the running configuration. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 3	switch(config)# **show mac access-lists** *name* **summary**	(Optional) Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.

	Command or Action	Purpose
		name—Specifies the ACL name. The maximum length is 64 characters.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example removes a MAC ACL:

```
switch# configure terminal
switch(config)# no mac access-list acl-mac-01
switch(config)# show mac access-lists acl-mac-01 summary
MAC ACL acl-mac-01
switch(config)# copy running-config startup-config
```

Changing Sequence Numbers in a MAC ACL

You can change sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
3. switch(config)# **resequence mac access-list** *name starting-sequence-number increment*
4. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) Displays the MAC ACL configuration for verification. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 3	switch(config)# **resequence mac access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by 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 number that you specify.

	Command or Action	Purpose
		name—Specifies the ACL name. The maximum length is 64 characters. *starting-sequence-number*—Specifies the sequence number of the first rule. The range is from 1 to 4294967295. *increment*—Specifies the amount by which to increment a sequence number to get the sequence number of each subsequent rule. The range is from 1 to 4294967295.
Step 4	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) Displays the MAC ACL configuration for verification. *name*—Specifies the ACL name. The maximum length is 64 characters.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example changes sequence numbers in a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
    10 permit 00c0.4f00.0000 0000.00ff.ffff any
    20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# resequence mac access-list acl-mac-01 100 10
switch(config)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
    100 permit 00c0.4f00.0000 0000.00ff.ffff any
    110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# copy running-config startup-config
```

Applying a MAC ACL as a Port ACL

You can apply a MAC ACL as a port ACL to any of the following interface types:

- Physical Ethernet interfaces
- Virtual Ethernet interface

A MAC ACL can also be applied to a port profile that is attached to a physical Ethernet interface or a virtual Ethernet interface.

ACLs cannot be applied on port channel interfaces. However, ACLs can be applied on a physical Ethernet interface that is not part of the port channel.

Before You Begin

- Log in to the CLI in EXEC mode.
- Ensure that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet** *slot-number* | **vethernet** *interface-number*}
3. switch(config-if)# **mac port access-group** *access-list* [**in** | **out**]
4. (Optional) switch(config-if)# **show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot-number* | **vethernet** *interface-number*}	Enters interface configuration mode for the specified interface. **ethernet—**Specifies the Ethernet IEEE 802.3z interface. *slot-number***—**Specifies the slot number. The range is from 1 to 514. **vethernet—**Specifies the virtual Ethernet interface. *interface-number***—**Specifies the interface number. The range is from 1 to 1048575.
Step 3	switch(config-if)# **mac port access-group** *access-list* [**in** | **out**]	Applies a MAC ACL to the interface. *access-list***—**Specifies the port ACL. The maximum length is 64 characters. **in—**Specifies inbound packets for the ACL. **out—**Specifies outbound packets for the ACL.
Step 4	switch(config-if)# **show running-config aclmgr**	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example applies a MAC ACL as a port ACL:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface ethernet 1
switch(config-if)# mac port access-group acl-mac-01 in
switch(config-if)#  show running-config aclmgr

!Command: show running-config aclmgr
!Time: Wed Mar 13 03:38:02 2013

version 5.2(1)SK1(2.1)
mac access-list acl-mac-01
    100 permit 00C0.4F00.0000 0000.00FF.FFFF any
    110 permit F866.F222.E5A6 FFFF.FFFF.FFFF any
```

```
interface Vethernet1
  mac port access-group acl-mac-01 in
switch(config-if)# copy running-config startup-config
```

Adding a MAC ACL to a Port Profile

Before You Begin

- Log in to the CLI in EXEC mode.
- Create the MAC ACL to add to this port profile and know its name.
- If you are using an existing port profile, know its name.
- If you are creating a new port profile, know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- Know the direction of packet flow for the access list.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. switch(config-port-prof)# **mac port access-group** *access-list* {**in** | **out**}
4. (Optional) switch(config-port-prof)# **show port-profile name** *profile-name*
5. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile. **type ethernet—**Specifies the Ethernet type for the port profile. **type vethernet—**Specifies the vEthernet type for the port profile. *name***—**Specifies the port profile. The maximum length is 80 characters.
Step 3	switch(config-port-prof)# **mac port access-group** *access-list* {**in** | **out**}	Adds the named ACL to the port profile for either inbound or outbound traffic. *access-list***—**Specifies the port ACL. The maximum length is 64 characters. **in—**Specifies inbound packets for the ACL. **out—**Specifies outbound packets for the ACL.
Step 4	switch(config-port-prof)# **show port-profile name** *profile-name*	(Optional) Displays the configuration for verification. *profile-name***—**Specifies the port profile name.

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

The following example adds a MAC ACL to a port profile

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# port-profile vm_eth1
switch(config-port-prof)# mac port access-group acl-mac-01 out
switch(config-port-prof)# show port-profile name vm_eth1

port-profile vm_eth1
type: Vethernet
description:
status: enabled
max-ports: 32
min-ports: 1
inherit:
config attributes:
mac port access-group acl-mac-01 out
no shutdown
evaluated config attributes:
mac port access-group acl-mac-01 out
no shutdown
assigned interfaces:
port-group: vm_eth1
system vlans: none
capability l3control: no
capability iscsi-multipath: no
capability vxlan: no
capability l3-vn-service: no
port-profile role: none
port-binding: static

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

Verifying MAC ACL Configurations

Use the following commands to verify the configuration:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.
show mac address-lists summary	Displays a summary of all configured MAC ACLs or a named MAC ACL.
show running-config aclmgr	Displays the ACL configuration, including MAC ACLs and the interfaces they are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL.

Monitoring MAC ACLs

Use the following commands for MAC ACL monitoring:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration. If the MAC ACL includes the **statistics per-entry** command, the **show mac access-lists** command output includes the number of packets that have matched each rule.
clear mac access-list counters	Clears statistics for all MAC ACLs or for a specific MAC ACL.

Configuration Examples for MAC ACLs

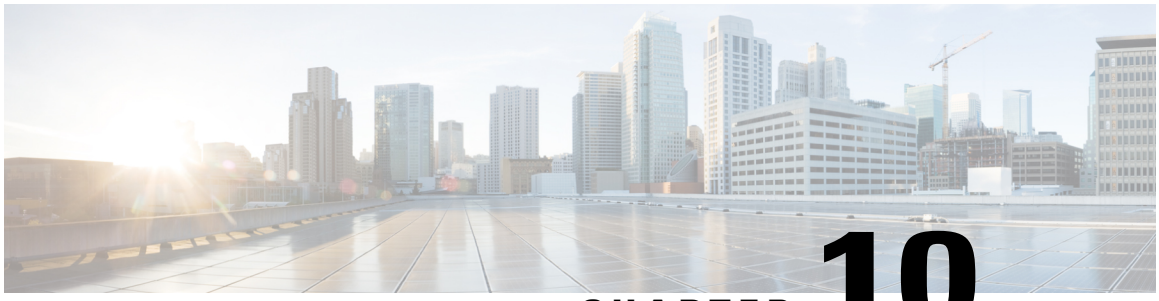
Configuration Example for Creating a MAC ACL for any Protocol

The following example shows how to create a MAC ACL named acl-mac-01, apply it as a port ACL on a physical Ethernet interface that is not a member of a port channel, and configure verification with match counters:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# 100 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# 110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# end
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface ethernet 3/5
switch(config-if)# mac port access-group acl-mac-01 out
switch(config-if)# show mac access-lists acl-mac-01 summary

MAC ACL acl-mac-01
    statistics per-entry
    Total ACEs Configured:2
    Configured on interfaces:
        Ethernet3/5 - egress (Port ACL)
    Active on interfaces:
        Ethernet3/5 - egress (Port ACL)
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
    statistics per-entry
    100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
    110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=546]
switch(config-if)# clear mac access-list counters
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
    statistics per-entry
    100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
    110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=0]
switch(config-if)#
```

Feature History for MAC ACLs

Feature Name	Releases	Feature Information
MAC ACL	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 10

Blocking Unknown Unicast Flooding

This chapter contains the following sections:

Information About UUFB

Unknown unicast packet flooding (UUFB) limits unknown unicast flooding in the forwarding path to prevent the security risk of unwanted traffic reaching the Virtual Machines (VMs). UUFB prevents packets received on both vEthernet and Ethernet interfaces destined to unknown unicast addresses from flooding the VLAN. When UUFB is applied, Virtual Ethernet Modules (VEMs) drop unknown unicast packets received on uplink ports, while unknown unicast packets received on vEthernet interfaces are sent out only on uplink ports.

Guidelines and Limitations for UUFB

- Before configuring UUFB, make sure that the VSM HA pair and all VEMs have been upgraded to the latest release by entering the **show module** command.
- You must explicitly disable UUFB on the ports of an application or VM by using MAC addresses other than the one given by .
- Unknown unicast packets are dropped by Cisco UCS fabric interconnects when Cisco UCS is running in end-host-mode.

- On Microsoft Network Load Balancing (MS-NLB) enabled vEthernet interfaces (by entering the **no mac auto-static-learn** command), UUFB does not block MS-NLB related packets. In these scenarios, UUFB can be used to limit flooding of MS-NLB packets to non-MS-NLB ports within a VLAN.

Default Settings for UUFB

Parameters	Default
uufb enable	Disabled
switchport uufb disable	Disabled

Configuring UUFB

Blocking Unknown Unicast Flooding Globally on the Switch

You can globally block unknown unicast packets from flooding the forwarding path for the switch.

Before You Begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **uufb enable**
3. (Optional) switch(config)# **show uufb status**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enables global configuration mode.
Step 2	switch(config)# [**no**] **uufb enable**	Configures UUFB globally for the VSM.
Step 3	switch(config)# **show uufb status**	(Optional) Displays the UUFB global setting for the VSM.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to block unknown unicast flooding globally:

```
switch# configure terminal
switch(config)# uufb enable
switch(config)# show uufb status
UUFB Status: Enabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Verifying the UUFB Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show uufb status	Displays the UUFB global setting for the VSM.
show running-config port-profile *profile-name*	Displays the running configuration for a specific port profile.
show running-config interface *vethernet interface-number*	Displays the running configuration for a specific interface.
vemcmd show port uufb-override	Displays UUFB disable state for each port.

Configuration Example for Blocking Unknown Unicast Packets

This example shows how to block unknown unicast packets from flooding the forwarding path globally for the VSM:

```
n1000v# config terminal
n1000v(config)# uufb enable
n1000v(config)# show uufb status
UUFB Status: Enabled
n1000v(config)# copy running-config startup-config
[########################################] 100%
```

Feature History for UUFB

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
UUFB	5.2(1)SK3(2.1)	This feature was introduced.