Cisco Nexus 3600 NX-OS Security Configuration Guide, Release 10.2(x)

First Published: 2021-08-23

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

Preface

This preface includes the following sections:

Audience

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

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 3000 Series Switches

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

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

Documentation Feedback

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

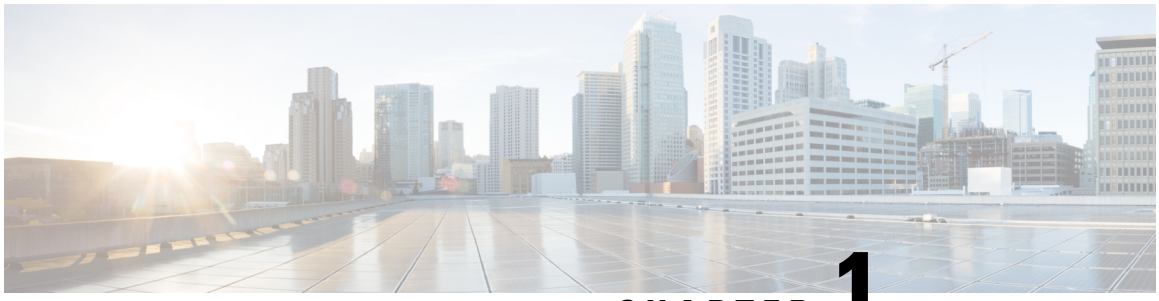
Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3600 Series NX-OS Security Configuration Guide, Release 10.2(x)*.

• New and Changed Information, on page 1

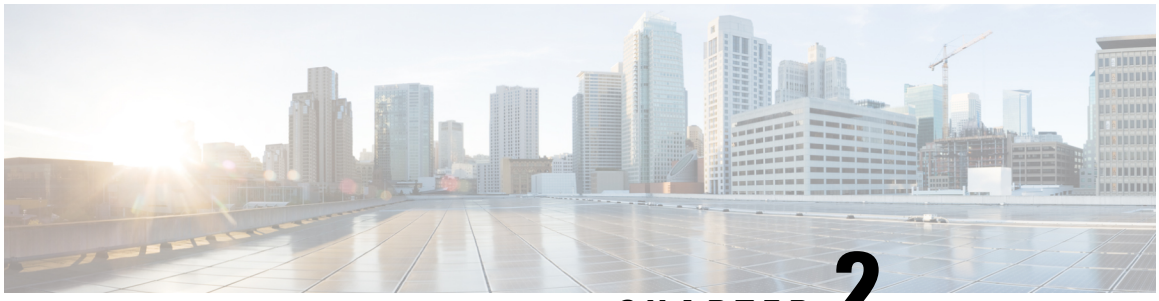
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3600 Series NX-OS Security Configuration Guide, Release 10.2(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 10.2(x)

Feature	Description	Changed in Release	Where Documented
No feature updates in this release		10.2(1)F	

New and Changed Information



CHAPTER 2

Overview

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

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Authentication, Authorization, and Accounting

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

Authentication

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

Authorization

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

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

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

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

RADIUS and TACACS+ Security Protocols

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

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

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

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

SSH and Telnet

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

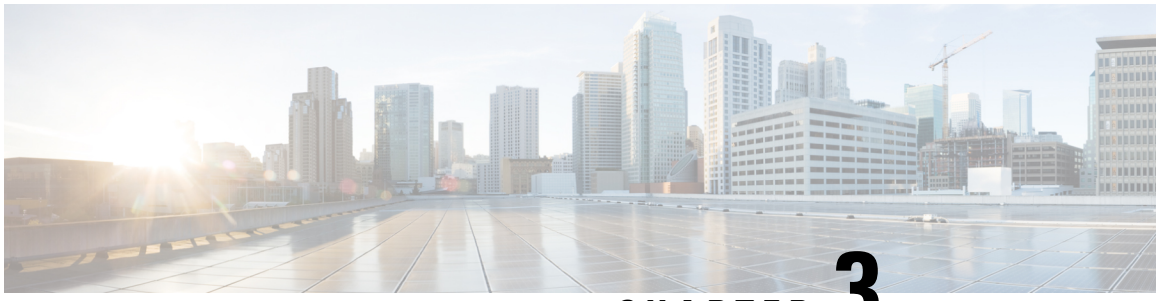
The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

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

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.



CHAPTER 3

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on Cisco NX-OS devices.

Information About AAA

AAA Security Services

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

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

AAA security provides the following services:

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

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

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

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

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA Services

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

- User password lists for each switch in the fabric are easier to manage.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- The accounting log for all switches in the fabric can be centrally managed.
- User attributes for each switch in the fabric are easier to manage than using the local databases on the switches.

AAA Server Groups

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

AAA Service Configuration Options

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

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication

- User management session accounting

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

Table 2: AAA Service Configuration Commands

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

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

- RADIUS server groups—Uses the global pool of RADIUS servers for authentication.
- Specified server groups—Uses specified RADIUS or TACACS+ server groups for authentication.
- Local—Uses the local username or password database for authentication.
- None—Uses only the username.

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

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

Table 3: AAA Authentication Methods for AAA Services

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

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

Authentication and Authorization Process for User Logins

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

- When you log in to the required Cisco Nexus device, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Cisco Nexus device sends an authentication request to the first AAA server in the group as follows:

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

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

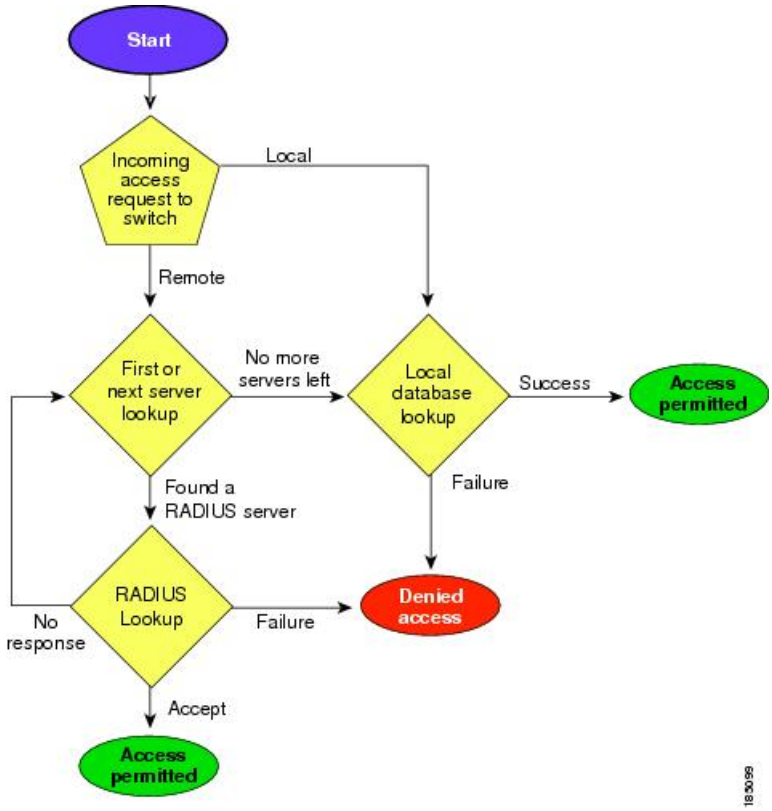
  If all configured methods fail, the local database is used for authentication.
- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

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

  If the AAA server protocol is TACACS+, another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If your username and password are successfully authenticated locally, the Cisco Nexus device logs you in and assigns you the roles configured in the local database.

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

Figure 1: Authentication and Authorization Flow for User Login



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

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable.
- The Cisco Nexus device is configured as a client of the AAA servers.
- The preshared secret key is configured on the Cisco Nexus device and on the remote AAA servers.
- The remote server responds to AAA requests from the Cisco Nexus device.

Guidelines and Limitations for AAA

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

If you configure the AAA login authentication default group, TACACS-SERVER-GROUP, it also overrides the login for the console. This override occurs even if **aaa authentication login console local** is a default command on the switch. To prevent this, you must configure **aaa authentication login console local**.

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

Configuring AAA

Configuring Console Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the Cisco Nexus device.
- Username only **none**

The default method is local.

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

Note If you configure the AAA login authentication default group, TACACS-SERVER-GROUP, it also overrides the login for the console. This override occurs even if **aaa authentication login console local** is a default command on the switch. To prevent this, you must configure **aaa authentication login console local**.

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

Procedure

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

Example

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

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

```
switch(config)# exit
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The default method is local.

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

Procedure

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

Enabling Login Authentication Failure Messages

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

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

Procedure

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

Logging Successful and Failed Login Attempts

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

Procedure

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

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

Configuring AAA Command Authorization

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

The authorization methods include the following:

- Group—TACACS+ server group
- Local—Local role-based authorization
- None—No authorization is performed

The default method is Local.

Note There is no authorization on the console session.

Before you begin

You must enable TACACS+ before configuring AAA command authorization.

Procedure

	Command or Action	Purpose

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

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

Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Enabling MSCHAP Authentication

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

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

The following table describes the RADIUS VSAs required for MSCHAP.

Table 4: MSCHAP RADIUS VSAs

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

Procedure

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

Configuring AAA Accounting Default Methods

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

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

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

- RADIUS server group—Uses the global pool of RADIUS servers for accounting.
- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.
- Local—Uses the local username or password database for accounting.

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

Before you begin

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

Procedure

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

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

About No Service Password-Recovery

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

Enabling No Service Password-Recovery

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

Step 2	**no service password-recovery**
	Example:
	`switch(config)# no service password-recovery`
	`WARNING: Executing this command will disable the password recovery mechanism. Do not execute this command without another plan for password recovery. Are you sure you want to continue? (y/n) : [y] y`
	`switch(config)# copy run start`
	`[########################################] 100%`
	`Copy complete, now saving to disk (please wait)...`
	`Copy complete.`
		Disables the password recovery mechanism.

	Command or Action	Purpose
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Step 4	**Reload**
	Example:
	`switch(config)# Reload`
	`This command will reboot the system. (y/n)?  [n] y`
	`2018 Jun 26 16:23:19 BAR %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command Line Interface`
	
	`CISCO SWITCH Ver 8.34`
	
	`CISCO SWITCH Ver 8.34`
	`Manual system restart from Command Line Interface`
	`writing reset reason 9,`
	`..`
	`..`
	
	`switch(boot)# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(boot)(config)# admin-password Abcd!123$`
	`ERROR: service password-recovery disabled. Cannot change password!`
	`switch(boot)(config)#`	

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

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

Using AAA Server VSAs

VSAs

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

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

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

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

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

VSA Format

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

- Shell— Used in access-accept packets to provide user profile information.
- Accounting—Used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Cisco Nexus device:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space.
- accountinginfo—Stores additional accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

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

```
shell:roles="roleA roleB …"
```

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

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

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

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

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

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

Secure Login Enhancements

Secure Login Enhancements

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

- Configuring Login Parameters
- Configuration Examples for Login Parameters
- Restricting Sessions Per User—Per User Per Login
- Enabling the Password Prompt for User Name
- Configuring Share Key Value for using RADIUS/TACACS+

Configuring Login Parameters

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

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Switch(config)# login block-for 100 attempts 2 within 100`	Configures your Cisco NX-OS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.

	Command or Action	Purpose
Step 3	**[no] login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** `Switch(config)# login quiet-mode access-class myacl`	(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.
Step 4	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show login failures** **Example:** `Switch# show login`	Displays login parameters. • **failures** --Displays information related only to failed login attempts.

Configuration Examples for Login Parameters

Setting Login Parameters Example

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

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

Showing Login Parameters Example

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

```
Switch# show login

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

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

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

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

```
Switch# show login failures
```

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

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

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

Restricting Sessions Per User—Per User Per Login

Use this task to restrict the maximum sessions per user.

Procedure

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

Enabling the Password Prompt for User Name

Procedure

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

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

Configuring Share Key Value for using RADIUS/TACACS+

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

Procedure

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

Monitoring and Clearing the Local AAA Accounting Log

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

Procedure

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

Verifying the AAA Configuration

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

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

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

Configuration Examples for AAA

The following example shows how to configure AAA:

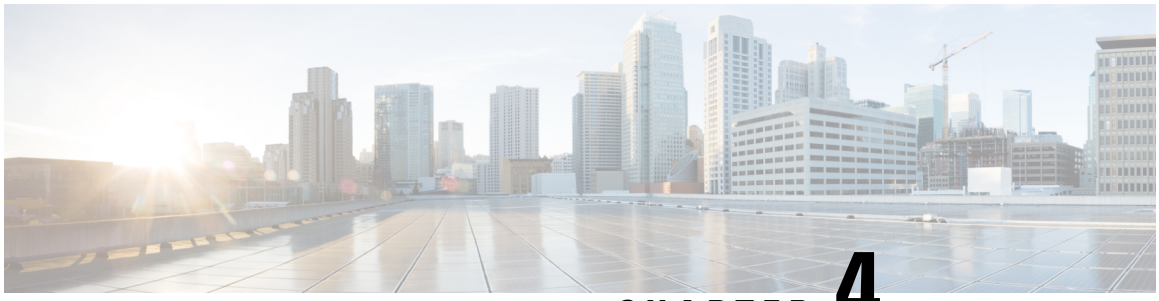
```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

Default AAA Settings

The following table lists the default settings for AAA parameters.

Table 5: Default AAA Parameters

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



CHAPTER 4

Configuring RADIUS

This chapter contains the following sections:

Information About RADIUS

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

RADIUS Network Environments

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

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

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

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

- Networks already using RADIUS.

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

- Networks that require resource accounting.

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

- Networks that support authentication profiles.

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

Information About RADIUS Operations

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

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   - ACCEPT—The user is authenticated.
   - REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

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

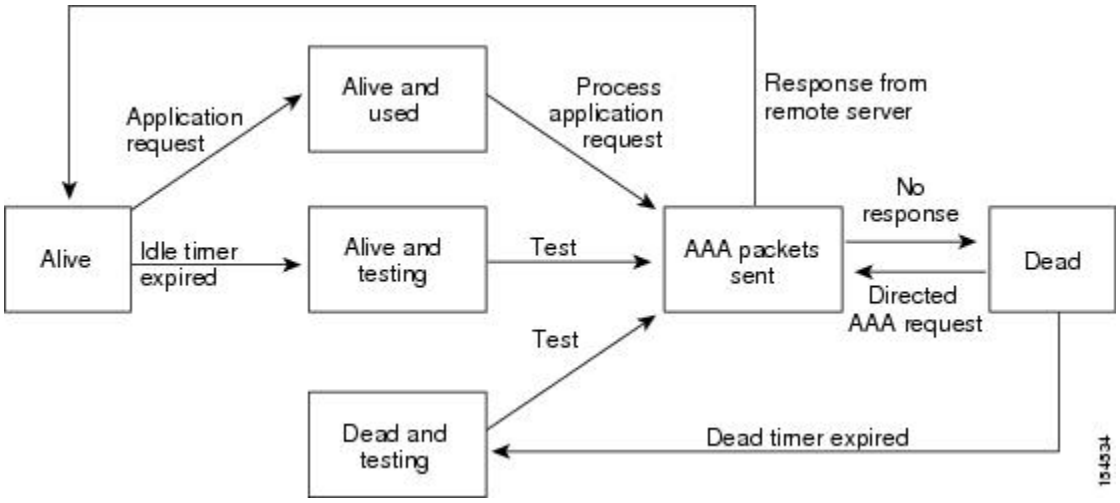
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

RADIUS Server Monitoring

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

The following figure shows the different RADIUS server states:

Figure 2: RADIUS Server States



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

Vendor-Specific Attributes

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

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

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

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

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

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

The Cisco Nexus device supports the following attributes:

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

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- You must obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.
- You must obtain preshared keys from the RADIUS servers.
- Ensure that the Cisco Nexus device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following configuration guidelines and limitations:

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

Configuring RADIUS Servers

This section describes how to configure RADIUS servers.

Procedure

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

See Configuring RADIUS Server Hosts, on page 33.

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

See Configuring RADIUS Global Preshared Keys, on page 33.

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

See Allowing Users to Specify a RADIUS Server at Login, on page 37 and Configuring Accounting and Authentication Attributes for RADIUS Servers, on page 39.

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

- Dead-time interval. See Configuring the Dead-Time Interval, on page 41.
- Allow specification of a RADIUS server at login. See Allowing Users to Specify a RADIUS Server at Login, on page 37

- Transmission retry count and timeout interval. See Configuring the Global RADIUS Transmission Retry Count and Timeout Interval, on page 38.
- Accounting and authentication attributes. See Configuring Accounting and Authentication Attributes for RADIUS Servers, on page 39.

Step 5 If needed, configure periodic RADIUS server monitoring.

See Configuring Periodic RADIUS Server Monitoring, on page 40.

Configuring RADIUS Server Hosts

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

Procedure

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

Example

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

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

Configuring RADIUS Global Preshared Keys

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

Before you begin

Obtain the preshared key values for the remote RADIUS servers

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **radius-server key** [**0** | **7**] *key-value*	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-contig**	Saves the change persistenetly through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

Configuring RADIUS Server Preshared Keys

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

Before you begin

Obtain the preshared key values for the remote RADIUS servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-contig**	Saves the change persistenetly through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS preshared keys:

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

Configuring RADIUS Server Groups

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

Procedure

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

	Command or Action	Purpose
Step 2	switch (config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive, alphanumeric string with a maximum of 127 characters.
Step 3	switch (config-radius)# **server** {*ipv4-address* | *ipv6-address* | *server-name*}	Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) switch (config-radius)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-radius)# **source-interface** *interface*	Assigns a source interface for a specific RADIUS server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip radius source-interface** command.
Step 6	switch(config-radius)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server group** [*group-name*]	Displays the RADIUS server group configuration.
Step 8	(Optional) switch(config)# **copy running-config startup-contig**	Saves the change persistenetly through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

```
switch# configure terminal
switch (config)# aaa group server radius RadServer
switch (config-radius)# server 10.10.1.1
switch (config-radius)# deadtime 30
```

```
switch (config-radius)# use-vrf management
switch (config-radius)# exit
switch (config)# show radius-server group
switch (config)# copy running-config startup-config
```

What to do next

Apply the RADIUS server groups to an AAA service.

Configuring the Global Source Interface for RADIUS Server Groups

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

Procedure

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

Example

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

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

Allowing Users to Specify a RADIUS Server at Login

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **radius-server directed-request**	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server directed-request**	Displays the directed request configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

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

Procedure

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

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

Example

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

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

Configuring Accounting and Authentication Attributes for RADIUS Servers

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

Procedure

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

	Command or Action	Purpose
Step 8	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

Configuring Periodic RADIUS Server Monitoring

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

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

	Command or Action	Purpose
Step 3	switch(config)# **radius-server deadtime** *minutes*	Specifies the number of minutes before the switch checks a RADIUS server that was previously unresponsive. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-contig**	Saves the change persistenetly through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

Configuring the Dead-Time Interval

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

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group. See Configuring RADIUS Server Groups, on page 35.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **radius-server deadtime**	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Manually Monitoring RADIUS Servers or Groups

Procedure

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

Example

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

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

Verifying the RADIUS Configuration

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

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

Displaying RADIUS Server Statistics

Procedure

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

Clearing RADIUS Server Statistics

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

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

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

Configuration Examples for RADIUS

The following example shows how to configure RADIUS:

```
switch# configure terminal
switch(config)# radius-server key 7 "ToIkLhPpG"
switch(config)# radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
```

```
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# exit
switch(config-radius)# use-vrf management
```

Default Settings for RADIUS

The following table lists the default settings for RADIUS parameters.

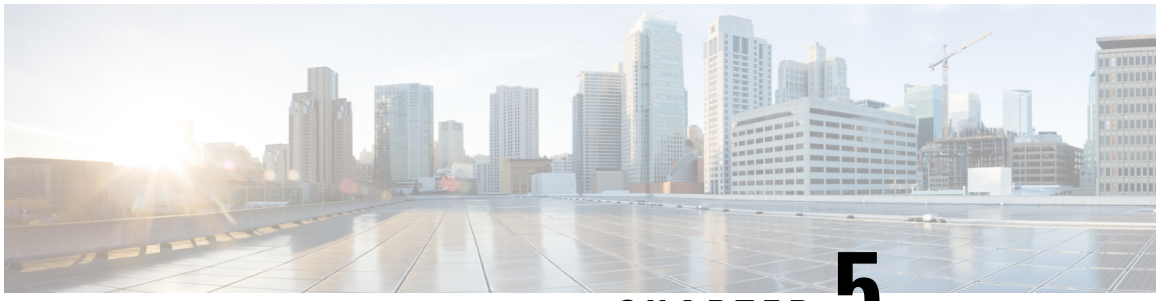
Table 6: Default RADIUS Parameters

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

Feature History for RADIUS

Table 7: Feature History for RADIUS

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



CHAPTER 5

Configuring TACACS+

This chapter contains the following sections:

Information About Configuring TACACS+

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

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

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

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Nexus device can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

User Login with TACACS+

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

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

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

2. The Cisco Nexus device receives one of the following responses from the TACACS+ daemon:
   - ACCEPT—User authentication succeeds and service begins. If the Cisco Nexus device requires user authorization, authorization begins.
   - REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   - ERROR—An error occurred at some time during authentication dither at the daemon or in the network connection between the daemon and the Cisco Nexus device. If the Cisco Nexus device receives an ERROR response, the switch tries to use an alternative method for authenticating the user.

   The user also undergoes an additional authorization phase, if authorization has been enabled on the Cisco Nexus device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the Cisco Nexus device again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

Services include the following:

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

Default TACACS+ Server Encryption Type and Preshared Key

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

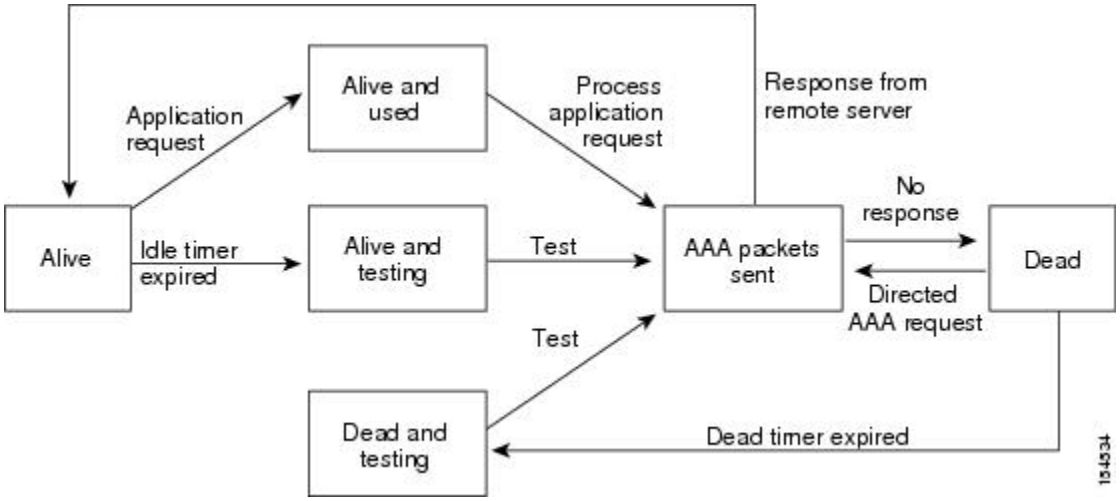
You can override the global preshared key assignment by using the **key** option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

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

The following figure shows the different TACACS+ server states:

Figure 3: TACACS+ Server States



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

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- You must obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.
- You must obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations for TACACS+

TACACS+ has the following configuration guidelines and limitations:

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

Configuring TACACS+

TACACS+ Server Configuration Process

This section describes how to configure TACACS+ servers.

Procedure

Step 1 Enable TACACS+.

See Enabling TACACS+ , on page 48.

Step 2 Establish the TACACS+ server connections to the Cisco Nexus device.

Configuring TACACS+ Server Hosts, on page 49

Step 3 Configure the preshared secret keys for the TACACS+ servers.

Configuring TACACS+ Global Preshared Keys, on page 49

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

Configuring TACACS+ Server Groups, on page 51

Step 5 If needed, configure periodic TACACS+ server monitoring.

Configuring Periodic TACACS+ Server Monitoring, on page 54

Enabling TACACS+

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature tacacs+**	Enables TACACS+.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IPv4 or IPv6 address or the hostname for the TACACS+ server on the Cisco Nexus device. All TACACS+ server hosts are added to the default TACACS+ server group.You can configure up to 64 TACACS+ servers.

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

(See Configuring TACACS+ Global Preshared Keys and Configuring TACACS+ Server Preshared Keys sections for more details.)

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

- Enable TACACS+. See Enabling TACACS+ , on page 48 for more information.
- Obtain the IPv4 or IPv6 addresses or the hostnames for the remote TACACS+ servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*}	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.
Step 3	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}	Specifies the IPv4 address or hostname for a TACACS+ server.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

Configuring TACACS+ Global Preshared Keys

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

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

- Enable TACACS+.
- Obtain the preshared key values for the remote TACACS+ servers.

Procedure

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

Step 2	**tacacs-server key** [**0** | **6** | **7**] *key-value*
	Example:
	`switch(config)# tacacs-server key 0 QsEfThUkO`
	Example:
	`switch(config)# tacacs-server key 7 "fewhg"`
		Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.
		By default, no secret key is configured.
		Note If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example.

Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure global preshared keys:

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

Configuring TACACS+ Server Groups

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

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

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa group server tacacs+** *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.
Step 3	(Optional) switch(config-tacacs+)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 4	(Optional) switch(config-tacacs+)# **source-interface** *interface*	Assigns a source interface for a specific TACACS+ server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip tacacs source-interface** command.
Step 5	switch(config-tacacs+)# **exit**	Exits configuration mode.
Step 6	(Optional) switch(config)# **show tacacs-server groups**	Displays the TACACS+ server group configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Configuring the Global Source Interface for TACACS+ Server Groups

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip tacacs source-interface** *interface* **Example:** `switch(config)# ip tacacs source-interface mgmt 0`	Configures the global source interface for all TACACS+ server groups configured on the device. The source interface can be the management or the VLAN interface.

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

Step 4	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Global TACACS+ Timeout Interval

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

Procedure

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

Configuring the Timeout Interval for a Server

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

Procedure

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

Configuring TCP Ports

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

Procedure

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

Example

The following example shows how to configure TCP ports:

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

Configuring Periodic TACACS+ Server Monitoring

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

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the number minutes before the Cisco Nexus device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes and the valid range is 0 to 1440 minutes.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Configuring the Dead-Time Interval

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

Note When the dead-time interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group. See Configuring TACACS+ Server Groups, on page 51

Procedure

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

Manually Monitoring TACACS+ Servers or Groups

Procedure

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

	Command or Action	Purpose
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a TACACS+ server group to confirm availability.

Example

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

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

Disabling TACACS+

You can disable TACACS+.

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

Procedure

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

Displaying TACACS+ Statistics

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

Procedure

	Command or Action	Purpose
Step 1	switch# **show tacacs-server statistics** [*hostname* | *ipv4-address* | *ipv6-address*]	Displays the TACACS+ statistics. **Note** *ipv6-address* parameter not supported on Nexus 3548.

Example

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

Verifying the TACACS+ Configuration

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

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

Configuration Examples for TACACS+

This example shows how to configure TACACS+:

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

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

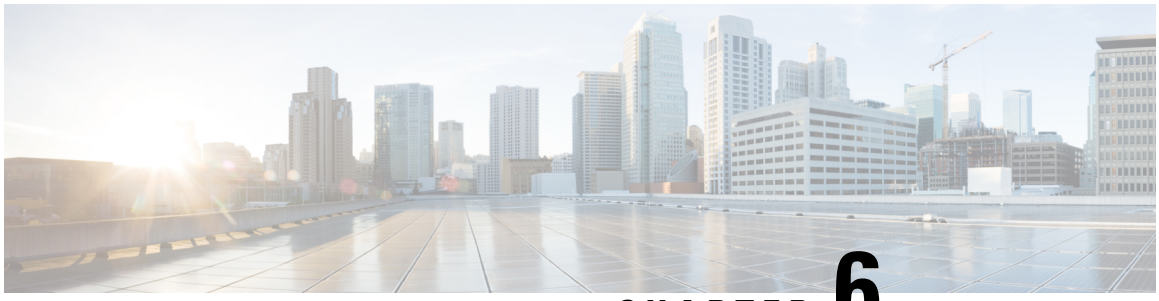
```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ikvhw10"
switch(config)# tacacs-server host 1.1.1.1
switch(config)# tacacs-server host 1.1.1.2

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

Default Settings for TACACS+

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

Table 8: Default TACACS+ Parameters

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



CHAPTER 6

Configuring SSH and Telnet

This chapter contains the following sections:

Information About SSH and Telnet

SSH Server

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

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

SSH Client

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

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

SSH Server Keys

SSH requires server keys for secure communications to the Cisco Nexus device. You can use SSH 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 Algorithm (DSA)

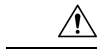
Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts three types of key-pairs for use by SSH version 2:

- 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, the Cisco Nexus device generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)



Caution If you delete all SSH keys, you can't start the SSH services.

SSH Authentication Using Digital Certificates

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

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

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

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

Telnet Server

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

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

Guidelines and Limitations for SSH

SSH has the following configuration guidelines and limitations:

- The Cisco Nexus device supports only SSH version 2 (SSHv2).
- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.

Configuring SSH

Generating SSH Server Keys

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration move.
Step 2	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key. The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show ssh key [dsa | rsa] [md5]**	Displays the SSH server keys. For Cisco NX-OS Release 7.0(3)I4(6) and any later 7.0(3)I4(x) release, this command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Specifying the SSH Public Keys for User Accounts

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

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

Specifying the SSH Public Keys in Open SSH Format

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

Procedure

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

Example

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

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

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

Specifying the SSH Public Keys in IETF SECSH Format

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

Procedure

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

Example

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

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP

	Command or Action	Purpose
Step 2	switch# **configure terminal**	Enters global configuration move.
Step 3	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Configuring the SSH Source Interface

You can configure SSH to use a specific interface.

Procedure

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

Example

This example shows how to configure the SSH source interface:

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

Starting SSH Sessions to Remote Devices

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

Procedure

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

Clearing SSH Hosts

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

Procedure

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

Disabling the SSH Server

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

Procedure

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

Deleting SSH Server Keys

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

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

Procedure

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

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

Procedure

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

Configuration Examples for SSH

The following example shows how to configure SSH:

Procedure

Step 1 Generate an SSH server key.

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

Step 2 Enable the SSH server.

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

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

Step 3 Display the SSH server key.

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

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

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

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

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
```

Step 5 Save the configuration.

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

Configuring X.509v3 Certificate-Based SSH Authentication

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

Before you begin

Enable the SSH server on the remote device.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**username** *user-id* [**password** [**0** | **5**] *password*] **Example:** `switch(config)# username jsmith password 4Ty18Rnt`	Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames. Usernames must begin with an alphanumeric character. The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device. **Note** If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.
Step 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**} **Example:** `switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`	Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.
Step 4	[**no**] **crypto ca trustpoint** *trustpoint* **Example:** `switch(config)# crypto ca trustpoint winca`	Configures a trustpoint.
Step 5	[**no**] **crypto ca authentication** *trustpoint* **Example:** `switch(config)# crypto ca authentication winca`	Configures a certificate chain for the trustpoint.
Step 6	**crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl**	Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot

	Command or Action	Purpose
	Example: `switch(config)# crypto ca crl request winca bootflash:crllist.crl`	of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA). **Note** Static CRL is the only supported revocation check method.
Step 7	(Optional) **show crypto ca certificates** **Example:** `switch(config)# show crypto ca certificates`	Displays the configured certificate chain and associated trustpoint.
Step 8	(Optional) **show crypto ca crl** *trustpoint* **Example:** `switch(config)# show crypto ca crl winca`	Displays the contents of the CRL list of the specified trustpoint.
Step 9	(Optional) **show user-account** **Example:** `switch(config)# show user-account`	Displays configured user account details.
Step 10	(Optional) **show users** **Example:** `switch(config)# show users`	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

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

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

show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
```

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

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

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

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

Configuring Telnet

Enabling the Telnet Server

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

Procedure

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

Reenabling the Telnet Server

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

Procedure

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

Configuring the Telnet Source Interface

You can configure Telnet to use a specific interface.

Procedure

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

Example

This example shows how to configure the Telnet source interface:

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

Starting Telnet Sessions to Remote Devices

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

- Obtain the hostname for the remote device and, if needed, obtain the username on the remote device.
- Enable the Telnet server on the Cisco Nexus device.
- Enable the Telnet server on the remote device.

Procedure

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

Example

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

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

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco Nexus device.

Procedure

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

Verifying the SSH and Telnet Configuration

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

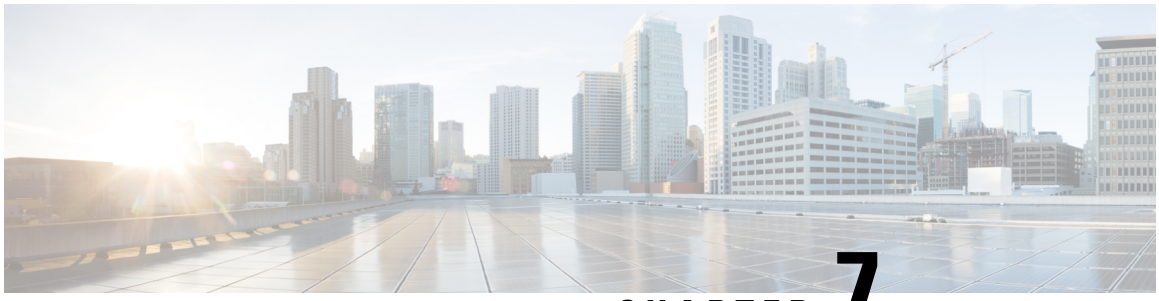
Command or Action	Purpose
switch# **show ssh key [dsa | rsa][md5]**	Displays SSH server keys.
switch# **show running-config security [all]**	Displays the SSH and user account configuration in the running configuration. The all keyword displays the default values for the SSH and user accounts.
switch# **show ssh server**	Displays the SSH server configuration.
switch# **show user-account**	Displays user account information.
switch# **show users**	Displays the users logged into the device.
switch# **show crypto ca certificates**	Displays the configured certificate chain and associated trustpoint for X.509v3 certificate-based SSH authentication.
switch# **show crypto ca crl** *trustpoint*	Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication.

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 9: Default SSH Parameters

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



CHAPTER 7

Configuring IP ACLs

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

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

Information About ACLs

An access control list (ACL) is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the switch determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether the packet is permitted or denied. If there is no match, the switch applies the applicable default rule. The switch continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

IP ACL Types and Applications

The Cisco Nexus device supports IPv4, IPv6, and MAC ACLs for security traffic filtering. The switch allows you to use IP access control lists (ACLs) as port ACLs, and Router ACLs as shown in the following table.

Table 10: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface	IPv4 ACLs IPv6 ACLs MAC ACLs
Router ACL	• Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Management interfaces • Switched Virtual Interfaces (SVIs)	IPv4 ACLs IPv6 ACLs
VTY ACL	VTYs	IPv4 ACLs IPv6 ACLs

Application Order

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

1. Port ACL
2. Ingress Router ACL

Rules

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The switch allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host.

Protocols

IPv4, IPv6, and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 ACL, you can specify ICMP by name.

You can specify any protocol by the integer that represents the Internet protocol number.

Implicit Rules

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the switch applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the switch denies unmatched IP traffic.

All IPv6 ACLs include the following implicit rule:

```
deny ipv6 any any
```

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. IPv4 ACLs support the following additional filtering options:

- Layer 4 protocol
- TCP and UDP ports
- IGMP types
- Established TCP connections

Sequence Numbers

The Cisco Nexus device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.
- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  switch(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, the device allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers.

The Cisco Nexus device stores operator-operand couples in registers called logical operation units (LOUs) to perform operations (greater than, less than, not equal to, and range) on the TCP and UDP ports specified in an IP ACL.

Prerequisites for ACLs

IP ACLs have the following prerequisites:

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

Guidelines and Limitations for ACLs

IP ACLs have the following configuration guidelines and limitations:

- As an enhancement to HTTP method match, the tcp-option-length option has been added to the ACE syntax to specify the length of the TCP options header in the packets. You can configure up to four tcp-option-lengths in the ACEs, which include the TCP option length of 0. If you do not configure the tcp-option-length option, the length is considered as 0. It means that only the packets without the TCP options header can match this ACE. This feature gives more flexibility in such a way that the HTTP method can be matched even on the packets that have the variable length TCP options header.
- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.
- You can configure any number of ACLs as long as TCAM space is available.
- Egress RACLs are not supported in Release 7.x. although the configuration may be allowed without an error or warning.
- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:

- Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
- IPv4 packets that have IP options (additional IP packet header fields following the destination address field).
- IPv6 packets that have extended IPv6 header fields.

- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds. Make sure that the time range is valid and in an active state.
- To use the **match-local-traffic** option for all inbound and outbound traffic, you must first enable the ACL in the software.
- For a Cisco N3K-C36180YC-R switch with configured egress RACLs, before upgrading from a 7.x release to a 9.x release, follow these steps to ensure the RACLs are maintained and the upgrade is completed without issue:

  1. Add TCAM entries for egress RACL using the **hardware access-list tcam region e-racl** command.
  2. Save the configuration and reload.
  3. Upgrade to a 9.x release.

  For more information about configuring TCAM regions, see ACL TCAM Regions, on page 93 and Configuring ACL TCAM Region Sizes, on page 101.
- Beginning Cisco NX-OS Release 9.3(2), you can configure a user-defined MAC address limit between the range of 16–256 for Cisco Nexus 36180YC-R and 3636C-R switches.
- In Cisco NX-OS Release 9.3(3), Cisco Nexus 3636C-R platform switches support the following for egress IPv6 RACLs:
  - Layer 4 Protocol
  - TCP flags
  - Fragment
  - ACL logs
- In Cisco NX-OS Release 9.3(3), Cisco Nexus 3636C-R platform switches do not support the following:
  - Egress atomic updates
  - Egress router ACL on external TCAM
  - Egress router ACL with UDF
  - Router ACL v6 counters for both egress and ingress
  - Egress and igress router ACL IPv6 with l4 ops
  - Egress router ACL on subinterface
  - Egress and ingress router ACL with IPv6 ICMP Type and Code

- IPv6 ingress router ACL with tcp-flag
- IPv4 router ACL with extra option

- In Cisco NX-OS Release 9.3(3), Cisco Nexus 3636C-R platform switches support the following for egress IPv4 RACLs:
  - TCP flags
  - ICMP Type and Code
  - ACL logs
- When you enable the counters for the ACL TCAM entries using the hardware profile acl-stats module xx command, the input discard field in the show interface is always zero. This limitation is applicable only to the Cisco Nexus 3600 platform switches with N3K-C3636C-R and N3K-C36180YC-R line cards.
- In Cisco NX-OS Release 9.3(5), IPv6 egress ACL supports the following on Cisco Nexus 3636C-R and 36180YC-R switches:
  - Layer 4 Protocol
  - TCP flags
  - Fragment
  - ACL logs
  - IPv6 header fields

Default ACL Settings

The following table lists the default settings for IP ACLs parameters.

Table 11: Default IP ACLs Parameters

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

The following table lists the default settings for MAC ACLs parameters.

Table 12: Default MAC ACLs Parameters

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

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 or IPv6 ACL on the device and add rules to it.

Before you begin

We recommend that you perform the ACL configuration using the Session Manager. This feature allows you to verify the ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **ip access-list** *name*
	• **ipv6 access-list** *name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3	[*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **Example:** `switch(config-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for the specific Cisco Nexus device.
Step 4	**statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	**hardware profile acl-stats module** *xx* **Example:**	Enables counters for the ACL TCAM entries on both, the internal and external TCAM.

	Command or Action	Purpose
	`switch(config-acl)# hardware profile acl-stats module 1`	**Note** This command is applicable only for Cisco Nexus 9500 platform switches with -R and -RX line cards and Cisco Nexus 3636C-R and 36180YC-R switches. VLAN and SVI statistics are lost when you enable the counters.
Step 6	**reload** **Example:** `switch(config)# reload`	Reloads the switch. **Note** The **reload** command is mandatory for the Cisco Nexus 3636C-R and 36180YC-R switches.
Step 7	Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 9	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
```

This example shows how to create an IPv6 ACL:

```
switch# configure terminal
switch(config)# ipv6 access-list acl-01-ipv6
switch(config-ipv6-acl)# permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
```

Configuring IPv4 ACL Logging

To configure the IPv4 ACL logging process, you first create the access list, then enable filtering of IPv4 traffic on an interface using the specified ACL, and finally configure the ACL logging process parameters.

Procedure

	Command or Action	Purpose

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

Step 2	**ip access-list** *name*
	Example:
	`switch(config)# ip access-list logging-test`
	`switch(config-acl)#`
		Creates an IPv4 ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3	{**permit** | **deny**} **ip** *source-address destination-address* **log** **Example:** `switch(config-acl)# permit ip any 10.30.30.0/24 log`	Creates an ACL rule that permits or denies IPv4 traffic matching its conditions. To enable the system to generate an informational logging message about each packet that matches the rule, you must include the **log** keyword. The *source-address* and *destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, or **any** to designate any address.

Step 4	**exit**
	Example:
	`switch(config-acl)# exit`
	`switch(config)#`
		Updates the configuration and exits IP ACL configuration mode.

Step 5	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 6	**ip access-group** *name* **in** **Example:** `switch(config-if)# ip access-group logging-test in`	Enables the filtering of IPv4 traffic on an interface using the specified ACL. You can apply an ACL to inbound traffic.

Step 7	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Updates the configuration and exits interface configuration mode.

Step 8	**logging ip access-list cache interval** *interval* **Example:**	Configures the log-update interval (in seconds) for the ACL logging process. The default value

	Command or Action	Purpose
	`switch(config)# logging ip access-list cache interval 490`	is 300 seconds. The range is from 5 to 86400 seconds.
Step 9	**logging ip access-list cache entries** *number-of-flows* **Example:** `switch(config)# logging ip access-list cache entries 8001`	Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576.
Step 10	**logging ip access-list cache threshold** *threshold* **Example:** `switch(config)# logging ip access-list cache threshold 490`	If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message.
Step 11	**logging ip access-list detailed** **Example:** `switch(config)# logging ip access-list detailed`	Enables the ACL name, the sequence number of ACE, action, ACL direction, ACL filter type, and the ACL applied interface are displayed in the output of the **show logging ip access-list cache** command.
Step 12	**hardware rate-limiter access-list-log** *packets* **Example:** `switch(config)# hardware rate-limiter access-list-log 200`	Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000. **Note** Cisco Nexus NX-OS 7.0(3)F3(1) does not support the **hardware rate-limiter access-list-log** command.
Step 13	**acllog match-log-level** *severity-level* **Example:** `switch(config)# acllog match-log-level 5`	Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging).
Step 14	(Optional) **show logging ip access-list cache** [**detail**] **Example:** `switch(config)# show logging ip access-list cache`	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces, and so on.

Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip** | **ipv6**}**ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 4	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 5	(Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 6	(Optional) switch#**show ip access-lists** *name*	Displays the IP ACL configuration.
Step 7	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 86

Removing an IP ACL

You can remove an IP ACL from the switch.

Before you remove an IP ACL from the switch, be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **no** {**ip** | **ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration. The removed IP ACL should not appear.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch# **show** {**ip** | **ipv6**} **access-lists** *name*	Displays the IP ACL configuration.
Step 3	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL to mgmt0

You can apply an IPv4 or IPv6 ACL to the management interface (mgmt0).

Before you begin

Ensure that the ACL that you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	Purpose

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

Step 2	**ip access-group** *access-list* {**in** | **out**} **Example:**	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction

	Command or Action	Purpose
	`switch(config-if)#ip access-group acl-120 out`	specified. You can apply one router ACL per direction.
Step 3	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

- Creating an IP ACL

Applying an IP ACL as a Port ACL

You can apply an IPv4 ACL to a physical Ethernet interface or a PortChannel. ACLs applied to these interface types are considered port ACLs.

Note Some configuration parameters when applied to an PortChannel are not reflected on the configuration of the member ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** [*chassis*/]*slot*/*port* | **port-channel** *channel-number*}	Enters interface configuration mode for the specified interface.
Step 3	(Optional) switch# **show running-config**	Displays the ACL configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 or or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces

- Layer 3 Ethernet port-channel interfaces and subinterfaces
- Management interfaces

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port*[**.** *number*] • switch(config)# **interface port-channel** *channel-number*[**.** *number*] • switch(config)# **interface mgmt** *port*	Enters configuration mode for the interface type that you specified.
Step 3	Enter one of the following commands: • switch(config-if)# **ip access-group** *access-list* {*in*} • switch(config-if)# **ipv6 traffic-filter** *access-list* {*in*}	Applies an IPv4 or IPv6 ACL to the layer 3 interface for traffic in the ingress direction.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring an Interface MAC Address and Limit

You can configure a static MAC address on SVI, Layer 3 interfaces, port channels, Layer 3 subinterfaces, and tunnel interfaces. You can also configure static MAC addresses on a range of ports and port channels. However, all ports must be in Layer 3. Even if one port in the range of ports is in Layer 2, the command is rejected and an error message is displayed.

By default, the maximum MAC addresses that can be configured on a switch is 16. However, you can change this limit and set it to a range of MAC addresses between 16 to 256.

On vPC enabled switches, the configured limit includes both, the locally configured user-defined MAC addresses plus the synced user-defined MAC addresses from the vPC peer.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **mac-address** *static router MAC address* **Example:** `switch(config-if)# mac-address 0019.D2D0.00AE`	Configures the interface MAC address. The no form of this command removes the configuration. You can enter the MAC address in any one of the four supported formats: • E.E.E • EE-EE-EE-EE-EE-EE • EE:EE:EE:EE:EE:EE • EEEE.EEEE.EEEE **Note** Do not enter any of the following invalid MAC addresses: • Null MAC address—0000.0000.0000 • Broadcast MAC address—FFFF.FFFF.FFFF • Multicast MAC address—0100.DAAA.ADDD

Step 4	(Optional) **show interface ethernet** *slot/port*
	Example:
	`switch(config-if)# show interface ethernet 2/1`
	`switch(config)#`
		Displays all information for the interface.

Step 5	**mac address-table limit** *16-256* **user-defined**
	Example:
	`switch(config)# mac address-table limit 200 user-defined`
	`switch(config)#`
		Configures the maximum number of MAC addresses that can be configured on a switch.

	Command or Action	Purpose
Step 6	(Optional) **show mac address-table limit user-defined** **Example:** `switch(config)# show mac address-table limit user-defined`	Displays the maximum number of MAC-addresses that can be configured on a switch.

Example

The following example shows how to configure an interface MAC address:

```
switch# configure terminal
switch(config)# interface ethernet 3/3
switch(config-if)# mac-address aaaa.bbbb.dddd
switch(config-if)# show interface ethernet 3/3
switch(config-if)#
switch(config)# mac address-table limit 100 user-defined
Warning: Configure the same User-Defined Mac Limit on the peer.
Warning: New Fhrp max group limit is 390
switch# show  mac address-table limit user-defined
User Defined Mac Limit: 100
FHRP Mac Limit: 390
==============
```

Configuring a UDF-Based MAC ACL

This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to MAC ACLs.

Beginning Cisco NX-OS Release 9.3(2), you can configure UDF-based MAC access lists (ACLs) for Cisco Nexus 36180YC-R and 3636C-R platform switches.

Procedure

	Command or Action	Purpose

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

Step 2	**udf** *udf-name offset-base offset length* **Example:** `switch(config)# udf pktoff10 packet-start 10 2`	Defines the UDF as follows: • *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name. • *offset-base*—Specifies the UDF offset base as follows: {**packet-start**}. • *offset*—Specifies the number of bytes offset from the offset base.

	Command or Action	Purpose
		• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs. You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name* } **Example:** `switch(config)# hardware access-list tcam region ing-ifacl qualify udf pktoff10`	Attaches the UDFs to the ing-ifacl TCAM region, which applies to IPv4 or IPv6 port ACLs. Up to 18 UDFs are supported. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure that enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes, on page 101. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Step 6	**mac access-list** *udf-acl*
	Example:
	`switch(config)# mac access-list udfacl`
	`switch(config-acl)#`
		Creates a MAC access control list (ACL) and enters MAC ACL configuration mode.

	Command or Action	Purpose
Step 7	**permit mac** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit mac any any udf pktoff10 0x1234 0xffff`	Configures the MAC ACL to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff. A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

Step 8	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		Enters interface configuration mode for a Layer 2 port-channel interface.

Step 9	**mac port access-group** *udf-access-list* **Example:** `switch(config-if)# mac port access-group udf-acl-01`	Applies the UDF-based MAC ACL to the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an ACL for IPv6 Extension Headers

This procedure applies only to the following devices:

- Cisco Nexus 9504 and 9508 modular chassis with these line cards: N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R
- Cisco Nexus 3600 Platform Switches (N3K-C36180YC-R and N3K-C3636C-R)

Beginning with Cisco NX-OS Release 9.3(7), if you configure an IPv6 ACL on the devices listed here, you must include a new rule for the disposition of IPv6 packets that include extension headers. For more information about IPv6 extension headers, see "Simplified IPv6 Packet Header" in NX-OS Release 9.3(x) or later of the *Cisco Nexus 3600 NX-OS Unicast Routing Configuration Guide*.

Note The permit or deny rule that you choose in this procedure is applied to any IPv6 packet with at least one extension header regardless of any other ACL rule that matches the packet's other fields.

Procedure

	Command or Action	Purpose

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

Step 2	**ipv6 access-list** *name*
	Example:
	`switch(config)# ipv6 access-list acl-01`
	`switch(config-acl)#`
		Creates the IPv6 ACL and enters ACL configuration mode.

Step 3	**extension-header** {**permit-all** | **deny-all**}
	Example:
	`switch(config-acl)# extension-header permit-all`
	`switch(config-acl)#`
		Choose the desired action for matched packets:
		• **permit-all** — Any IPv6 packet with at least one extension header is permitted.
		• **deny-all** — Any IPv6 packet with at least one extension header is dropped.

About System ACLs

You can configure system ACLs on Cisco Nexus 36180YC-R and C3636C-R switches. With system ACLs, you can now configure a Layer 2 port ACL (PACL) on all the ports with the same access-list in the switch. Configuring system ACLs reduces the TCAM usage and also brings down the time and memory usage while the policy is being applied or modified.

See the following guidelines and limitations for configuring system ACLs:

- The system PACL is supported for Layer 2 interface only.
- ACE statistics are not yet supported for the system ACLs.
- IPv6 is not yet supported in the system ACLs.
- System ACLs are not supported on the breakout port.
- For quality of service, ACL, or TCAM carving configuration on Cisco Nexus 3600 platform switches, see the Cisco Nexus 3600 NX-OS Quality of Service Configuration Guide for more information.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

The IPv4 TCAMs are single wide.

You can create IPv6 port ACLs, router ACLs, and you can match IPv6 addresses for QoS. Cisco NX-OS provides simultaneous support for all three TCAMs. You must remove or reduce the size of the existing TCAMs to enable these new IPv6 TCAMs.

TCAM region sizes have the following guidelines and limitations:

- To revert to the default ACL TCAM size, use the **no hardware access list tcam region** command. You need to reload the modules when you revert to default sizes.
- Depending upon the platform, each TCAM region might have a different minimum/maximum/aggregate size restriction.
- The total number of TCAMs is 16.
  - There are 12 large TCAMs—Each has 2048 entries that are 160 bit key size.
  - There are 4 small TCAMs—Each has 256 entries that are 160 bit key size.
- The TCAM regions RACL v6, QoS, CoPP, and Multicast cannot be set to 0.
- Redirect_v6, RACL v4 cannot share TCAM with any other features.
- After TCAM carving, you must reload the switch.
- RACL v6, CoPP, and multicast have default TCAM sizes and these TCAM sizes must be non-zero on the following Cisco 3600 line cards to avoid line card failure during reload:
  - N3K-C3636C-R
  - N3K-C36180YC-R
- You can partially use IPv6 RACL with IPv6 IFCAL. This is applicable Cisco Nexus N3K-C36180YC-R and N3K-C3636C-R line cards.

Table 13: TCAM Sizes by ACL Region

TCAM ACL Region	Default Size
PACL_IPv4 [ifacl]	1024
PACL_IPV6 [ipv6-ifacl]	1024
PACL_MAC [mac-ifacl]	2048
IPv4 Port QOS [qos]	640
IPv6 Port QOS [ipv6-qos]	256
IPv4 RACL [racl]	1024
IPv6 RACL [ipv6-racl]	1024
IPv4 L3 QoS [l3qos]	640
IPv6 L3 QoS [ipv6-l3qos]	256
SPAN [span]	96
Ingress COPP [copp]	128
Redirect v4	1024
Redirect v6	2048

Carving a TCAM Region

Before configuring the system ACLs, carve the TCAM region first. Note that for configuring the ACLs less than 1k, you do not need to carve the TCAM region. See the Configuring ACL TCAM Region Sizes, on page 101 section for more information.

Note You can configure PACL IPv4, RACL IPv4, and RACL IPv6 beyond 12k.

Configuring System ACLs

After an IPv4 ACL is created, configure the system ACL.

Before you begin

Create an IPv4 ACL on the device. See Creating an IP ACL, on page 81 for more information.

Procedure

	Command or Action	Purpose
Step 1	**config t**	Enters the configuration mode.
Step 2	**system acl**	Configures the system ACL.
Step 3	**ip port access-group** <*pacl name*> **in**	Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.

Configuration and Show Command Examples for the System ACLs

See the following configuration examples for the system ACL show commands.

Configuring system PACL with 1K scale [using default TCAM]

See the following example for configuring system PACL with 1K scale [Using default TCAM].

Step 1: Create PACL.

```
config t
ip access-list PACL-DNA
    10 permit ip 1.1.1.1/32 any
    20 permit tcp 3.0.0.0/8 255.0.0.0 eq 1500
    25 deny udp any any eq 500
    26 deny tcp any eq 490 any
    …... ….

    1000 deny any any
```

Step 2: Apply PACL into system level.

```
configuration terminal
```

```
system acl
    ip port access-group PACL-DNA in
```

To validate the system ACLs that are configured on the switch, use the **sh run aclmgr | sec system** command:

```
switch# sh run aclmgr | sec system
system acl
 ip port access-group test in
switch#
```

To validate the PACLs that are configured on the switch, use the **sh ip access-lists <name> [summary]** command:

```
switch# sh ip access-lists test

IP access list test
        10 deny udp any any eq 27
        20 permit ip 1.1.1.1/32 100.100.100.100/32
        30 permit ip 1.2.1.1/32 100.100.100.100/32
        40 permit ip 1.3.1.1/32 100.100.100.100/32
        50 permit ip 1.4.1.1/32 100.100.100.100/32
        60 permit ip 1.5.1.1/32 100.100.100.100/32
        70 permit ip 1.6.1.1/32 100.100.100.100/32
        80 permit ip 1.7.1.1/32 100.100.100.100/32
        90 permit ip 1.8.1.1/32 100.100.100.100/32

switch# sh ip access-lists test summary
IPV4 ACL test
        Total ACEs Configured: 12279
        Configured on interfaces:
        Active on interfaces:
                - ingress
                - ingress

switch#
```

To validate PACL IPv4 (ifacl) TCAM region size, use the **show hardware access-list tcam region** command:

```
switch# show hardware access-list tcam region
*********************************WARNING*********************************
*****************The output shows NFE tcam region info*****************
***Please refer to 'show hardware access-list tcam template' for NFE2***
*************************************************************************
                             IPV4 PACL [ifacl] size = 12280
                          IPV6 PACL [ipv6-ifacl] size =    0
                            MAC PACL [mac-ifacl] size =    0
                             IPV4 Port QoS [qos] size =  640
                          IPV6 Port QoS [ipv6-qos] size =  256
                            MAC Port QoS [mac-qos] size =    0
                       FEX IPV4 PACL [fex-ifacl] size =    0
                    FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
                      FEX MAC PACL [fex-mac-ifacl] size =    0
                       FEX IPV4 Port QoS [fex-qos] size =    0
                  FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
                    FEX MAC Port QoS [fex-mac-qos] size =    0
                                 IPV4 VACL [vacl] size =    0
                              IPV6 VACL [ipv6-vacl] size =    0
                                MAC VACL [mac-vacl] size =    0
                               IPV4 VLAN QoS [vqos] size =    0
                          IPV6 VLAN QoS [ipv6-vqos] size =    0
```

```
                        MAC VLAN QoS [mac-vqos] size =    0
                               IPV4 RACL [racl] size =    0
                             IPV6 RACL [ipv6-racl] size =  128
                    IPV4 Port QoS Lite [qos-lite] size =    0
              FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
                   IPV4 VLAN QoS Lite [vqos-lite] size =    0
                    IPV4 L3 QoS Lite [l3qos-lite] size =    0
                          Egress IPV4 QoS [e-qos] size =    0
                     Egress IPV6 QoS [e-ipv6-qos] size =    0
                       Egress MAC QoS [e-mac-qos] size =    0
                           Egress IPV4 VACL [vacl] size =    0
                     Egress IPV6 VACL [ipv6-vacl] size =    0
                       Egress MAC VACL [mac-vacl] size =    0
                        Egress IPV4 RACL [e-racl] size =    0
                   Egress IPV6 RACL [e-ipv6-racl] size =    0
               Egress IPV4 QoS Lite [e-qos-lite] size =    0
                               IPV4 L3 QoS [l3qos] size =  640
                         IPV6 L3 QoS [ipv6-l3qos] size =  256
                           MAC L3 QoS [mac-l3qos] size =    0
                                    Ingress System size =    0
                                     Egress System size =    0
                                       SPAN [span] size =   96
                                  Ingress COPP [copp] size =  128
                        Ingress Flow Counters [flow] size =    0
switch#
```

To view ACL related tech support information, use the **show tech-support aclmgr** and **show tech-support aclqos** commands.

```
show tech-support aclmgr
show tech-support aclqos
```

Configuring ACL Logging

ACL Logging

The Cisco Nexus device supports ACL logging, which allows you to monitor flows that hit specific access control lists (ACLs). To enable the feature for the ACL entry, configure specific ACEs with the optional **log** keyword.

Configuring the ACL Logging Cache

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **logging ip access-list cache entries** *num_entries*	Sets the maximum number of log entries cached in the software. The range is from 0 to 1000000 entries. The default value is 8000 entries.

	Command or Action	Purpose
Step 3	switch(config)# **logging ip access-list cache interval** *seconds*	Sets the number of seconds between log updates. If an entry is inactive for this duration, it is removed from the cache. The range is from 5 to 86400 seconds. The default value is 300 seconds.
Step 4	switch(config)# **logging ip access-list cache threshold** *num_packets*	Sets the number of packet matches before an entry is logged. The range is from 0 to 1000000 packets. The default value is 0 packets, which means that logging is not triggered by the number of packet matches.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example show how to set the maximum number of log entries to 5000, the interval to 120 seconds, and the threshold to 500000:

```
switch# configure terminal
switch(config)# logging ip access-list cache entries 5000
switch(config)# logging ip access-list cache interval 120
switch(config)# logging ip access-list cache threshold 500000
switch(config)# copy running-config startup-config
```

Applying ACL Logging to an Interface

You can apply ACL logging to Ethernet interfaces and port channels.

Before you begin

- Create an ACL.
- Create an IP access list with at least one access control entry (ACE) configured for logging.
- Configure the ACL logging cache.
- Configure the ACL log match level.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies the Ethernet interface.
Step 3	switch(config-if)# **ip access-group** *name* **in**	Attaches an ACL with a log to the specified interface. ACL logging is enabled when the ACL is applied to the interface on the hardware.

	Command or Action	Purpose
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to apply the Ethernet interface with the logging specified in acl1 for all ingress traffic:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip access-group acl1 in
switch(config-if)# copy running-config startup-config
```

Applying the ACL Log Match Level

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **acllog match-log-level** *number*	Specifies the logging level to match for entries to be logged in the ACL log (acllog). The number is a value from 0 to 7. The default is 6. **Note** Log messages are entered into the log if the logging level for the ACL log facility (acllog) and the logging severity level for the log file are greater than or equal to the ACL log match log level setting.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to apply the log match level for entries to be logged in the ACL log:

```
switch# configure terminal
switch(config)# acllog match-log-level 3
switch(config)# copy running-config startup-config
```

Clearing Log Files

You can clear messages in the log file and the NVRAM.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear logging ip access-list cache**	Clears the access control list (ACL) cache.

Verifying the ACL Logging Configuration

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

Command	Purpose
switch# **show hardware access-list tcam region**	Displays the TCAM sizes that will be applicable on the next reload of the device.
switch# **show ip access-lists**	Displays the IPv4 ACL configuration.
switch# **show ipv6 access-lists**	Displays the IPv6 ACL configuration.
switch# **show logging ip access-list cache [detail]**	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, and source interfaces.
switch# **show logging ip access-list status**	Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value..
switch# **show startup-config acllog**	Displays the access control list (ACL) log file in the startup configuration.
switch# **show startup-config aclmgr [all]**	Displays the access control list (ACL) log file in the startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The all option displays both the default (CoPP-configured) and the user-configured ACLs in the startup configuration.
switch# **show running-config acllog**	Displays the access control list (ACL) log file in the running configuration.

Command	Purpose
switch# **show running-config aclmgr [all]**	Displays the access control list (ACL) log file in the running configuration including the IP ACL configuration and the interfaces where you have applied IP ACLs. **Note** This command displays the user-configured ACLs in the running configuration. The all option displays both the default (CoPP-configured) and the user-configured ACLs in the startup configuration.

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Note You cannot change the size of the small TCAMs (TCAM 12 through 15)

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**hardware access-list tcam region** { **ifacl** | {**ipv6-qos** | **qos**} | {**ipv6-racl** | **racl**} *tcam_size*	Changes the ACL TCAM region size. • **ifacl**—Configures the size of the interface ACL (ifacl) TCAM region. The maximum number of entries is 1500. • **qos**—Configures the size of the quality of service (QoS) TCAM region. • **racl**—Configures the size of the router ACL (RACL) TCAM region. • *tcam_size*—TCAM size. The range is from 0 to 256, 512, (multiples of 256) entries.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

	Command or Action	Purpose
Step 4	switch(config)# **show hardware access-list tcam region** **Example:** `switch(config)# show hardware access-list tcam region`	Displays the TCAM sizes that will be applicable on the next reload of the switch.
Step 5	switch(config)# **reload** **Example:** `switch(config)# reload`	Copies the running configuration to the startup configuration. **Note** The new size values are effective only upon the next reload after saving the **copy running-config to startup-config**.

Example

The following example shows how to change the size of the RACL TCAM region:

```
switch(config)# hardware access-list tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware accesslist tcam region | exclude "0"



        IPV4 PACL [ifacl] size = 1024
   IPV6 PACL [ipv6-ifacl] size = 1024
     MAC PACL [mac-ifacl] size = 2048
      IPV4 Port QoS [qos] size =  640
 IPV6 Port QoS [ipv6-qos] size =  256
         IPV4 RACL [racl] size = 2048
    IPV6 RACL [ipv6-racl] size = 1024
      IPV4 L3 QoS [l3qos] size =  640
 IPV6 L3 QoS [ipv6-l3qos] size =  256
              SPAN [span] size =   96
      Ingress COPP [copp] size =  128
              Redirect v4 size =  1024
              Redirect v6 size =  2048
```

Reverting to the Default TCAM Region Sizes

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **ipsg** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Reverts the configuration to the default ACL TCAM size.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **reload**	Reloads the switch.

Example

The following example shows how to revert to the default RACL TCAM region sizes:

```
switch(config)# no hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configuring ACLs on Virtual Terminal Lines

To restrict incoming and outgoing connections for IPv4 or IPv6 between a Virtual Terminal (VTY) line and the addresses in an access list, use the **access-class** command in line configuration mode. To remove access restrictions, use the **no** form of this command.

Follow these guidelines when configuring ACLs on VTY lines:

- Set identical restrictions on all VTY lines because a user can connect to any of them.
- Statistics per entry is not supported for ACLs on VTY lines.

Before you begin

Be sure that the ACL that you want to apply exists and is configured to filter traffic for this application.

Procedure

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

	Command or Action	Purpose

Step 2	switch(config)# **line vty**
	Example:
	`switch(config)# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

Step 3	switch(config-line)# **access-class access-list-number {in | out}**
	Example:
	`switch(config-line)# access-class ozi2 in`
	`switch(config-line)#access-class ozi3 out`
	`switch(config)#`
		Specifies inbound or outbound access restrictions.

Step 4	(Optional) switch(config-line)# **no access-class access-list-number {in | out}**
	Example:
	`switch(config-line)# no access-class ozi2 in`
	`switch(config-line)# no access-class ozi3 out`
	`switch(config)#`
		Removes inbound or outbound access restrictions.

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

Step 6	(Optional) switch# **show running-config aclmgr** **Example:** `switch# show running-config aclmgr`	Displays the running configuration of the ACLs on the switch.
Step 7	(Optional) switch# **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

The following example shows how to apply the access-class ozi2 command to the in-direction of the vty line.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# line vty
switch(config-line)# access-class ozi2 in
```

```
switch(config-line)# exit
switch#
```

Verifying ACLs on VTY Lines

To display the ACL configurations on VTY lines, perform one of the following tasks:

Command	Purpose
show running-config aclmgr	Displays the running configuration of the ACLs configured on the switch.
show users	Displays the users that are connected.
show access-lists *access-list-name*	Display the statistics per entry.

Configuration Examples for ACLs on VTY Lines

The following example shows the connected users on the console line (ttyS0) and the VTY lines (pts/0 and pts/1).

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    ttyS0        Aug 27 20:45   .         14425 *
admin    pts/0        Aug 27 20:06 00:46       14176 (172.18.217.82) session=ssh
admin    pts/1        Aug 27 20:52   .         14584 (10.55.144.118)
```

The following example shows how to allow vty connections to all IPv4 hosts except 172.18.217.82 and how to deny vty connections to any IPv4 host except 10.55.144.118, 172.18.217.79, 172.18.217.82, 172.18.217.92:

```
switch# show running-config aclmgr
!Time: Fri Aug 27 22:01:09 2010
version 5.0(2)N1(1)
ip access-list ozi
  10 deny ip 172.18.217.82/32 any
  20 permit ip any any
ip access-list ozi2
  10 permit ip 10.55.144.118/32 any
  20 permit ip 172.18.217.79/32 any
  30 permit ip 172.18.217.82/32 any
  40 permit ip 172.18.217.92/32 any


line vty
  access-class ozi in
  access-class ozi2 out
```

The following example shows how to configure the IP access list by enabling per-entry statistics for the ACL:

```
switch# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
switch(config)# ip access-list ozi2
switch(config-acl)# statistics per-entry
switch(config-acl)# deny tcp 172.18.217.83/32 any
switch(config-acl)# exit

switch(config)# ip access-list ozi
switch(config-acl)# statistics per-entry
```

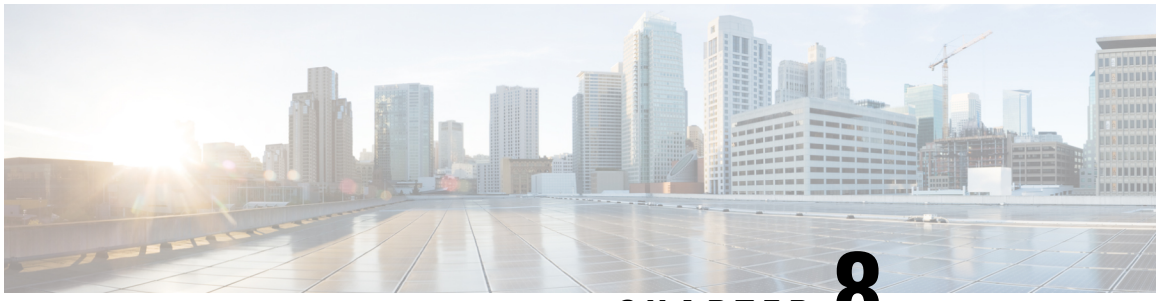
```
switch(config-acl)# permit ip 172.18.217.20/24 any
switch(config-acl)# exit
switch#
```

The following example shows how to apply the ACLs on VTY in and out directions:

```
switch(config)# line vty
switch(config-line)# ip access-class ozi in
switch(config-line)# access-class ozi2 out
switch(config-line)# exit
switch#
```

The following example shows how to remove the access restrictions on the VTY line:

```
switch# configure terminal
Enter configuration commands, one per line. End
with CNTL/Z.
switch(config)# line vty
switch(config-line)# no access-class ozi2 in
switch(config-line)# no ip access-class ozi2 in
switch(config-line)# exit
switch#
```



CHAPTER 8

Configuring Unicast RPF

This chapter contains the following sections:

Information About Unicast RPF

The Unicast RPF feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IPv4 source addresses into a network by discarding IPv4 packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IPv4 or IPv6 addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

When you enable Unicast RPF on an interface, the examines all ingress packets received on that interface to ensure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This examination of source addresses relies on the Forwarding Information Base (FIB).

Unicast RPF verifies that any packet received at a interface arrives on the best return path (return route) to the source of the packet by doing a reverse lookup in the FIB. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, the source address might have been modified by the attacker. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

Note With Unicast RPF, all equal-cost “best” return paths are considered valid, which means that Unicast RPF works where multiple return paths exist, if each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are being used and unequal candidate paths back to the source IP address exist.

Unicast RPF Process

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB that matches the route to the receiving interface. Static routes, network statements, and dynamic routing add routes to the FIB.
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

You can use Unicast RPF for downstream networks, even if the downstream network has other connections to the Internet.

Caution Be careful when using optional BGP attributes, such as weight and local preference, because an attacker can modify the best path back to the source address. Modification would affect the operation of Unicast RPF.

When a packet is received at the interface where you have configured Unicast RPF and ACLs, the Cisco NX-OS software performs the following actions:

Procedure

Step 1 Checks the input ACLs on the inbound interface.

Step 2 Uses Unicast RPF to verify that the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 Conducts a FIB lookup for packet forwarding.

Step 4 Checks the output ACLs on the outbound interface.

Step 5 Forwards the packet.

Global Statistics

Each time the Cisco NX-OS device drops a packet at an interface due to a failed unicast RPF check, that information is counted globally on the device on a per-forwarding engine (FE) basis. Global statistics on dropped packets provide information about potential attacks on the network, but they do not specify which interface is the source of the attack. Per-interface statistics on packets dropped due to a failed unicast RPF check are not available.

Guidelines and Limitations for Unicast RPF

Unicast RPF (uRPF) has the following configuration guidelines and limitations:

- You must apply uRPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream that you apply uRPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying uRPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying uRPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying uRPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy uRPF across Internet, intranet, and extranet resources, means that the better the chances are of mitigating large-scale network disruptions throughout the Internet community, and the better the chances are of tracing the source of an attack.
- uRPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure uRPF at a home gateway so that uRPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use uRPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.
- Do not use uRPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure uRPF only where there is natural or configured symmetry. Do not configure strict uRPF.
- uRPF allows packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.
- When uRPF is enabled, loose mode is applied for both IPv4 and IPv6. However, strict mode can be applied per protocol.
- For strict uRPF to work, you must enable it on both the ingress interface and the interface where the source IP address is learned.
- The switch hardware does not implement strict uRPF per the configured routing interface.
- Strict uRPF is implemented per learned route on strict uRPF-enabled interfaces.
- If a route is resolved as ECMP, strict uRPF will fall back to loose mode.
- Because of the hardware limitation on the trap resolution, uRPF might not be applied on supervisor-bound packets via inband.
- For IP traffic, both IPv4 and IPv6 configurations should be enabled simultaneously.
- Due to hardware limitations, the Cisco Nexus 3600 Series switches support only the following combinations:

uRPF Configuration		**Applied Traffic Check on Source IP Address**		
IPv4	IPv6	IP Unipath	IP ECMP	MPLS Encap/VPN/ECMP
Disable	Disable	Allow	Allow	Allow

uRPF Configuration		Applied Traffic Check on Source IP Address		
Loose	Loose	uRPF loose	uRPF loose	uRPF loose
Strict	Strict	uRPF strict	uRPF loose	uRPF loose

Default Settings for Unicast RPF

This table lists the default settings for Unicast RPF parameters.

Table 14: Default Unicast RPF Parameter Settings

Parameters	Default
Unicast RPF	Disabled

Configuring Unicast RPF

You can configure the Strict Unicast RPF mode or the Loose Unicast RPF mode on the ingress interface. For Strict Unicast mode, apply the configuration to interfaces where the source IP is attached. This allows you to configure the allowed list of specific sources.

Strict Unicast RPF mode
A strict mode check is successful when Unicast RPF finds a match in the FIB for the packet source address and the ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match. If this check fails, the packet is discarded. You can use this type of Unicast RPF check where packet flows are expected to be symmetrical.

Loose Unicast RPF mode
A loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet is received isn't required to match any of the interfaces in the FIB result.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Specifies an ethernet interface and enters interface configuration mode.

	Command or Action	Purpose
Step 3	{**ip** | **ipv6**} **verify unicast source reachable-via any** **Example:** `switch(config-if)# ip verify unicast source reachable-via any`	Configures unicast RPF on the interface for both IPv4 and IPv6. **Note** Configure unicast RPF on each interface, since it's disabled by default. The configuration is shared across both IPv4 and IPv6. If you enable or disable on either IPv4 and IPv6, it affects all protocols on that interface. **Note** When you enable uRPF for IPv4 or IPv6 (using the **ip** or **ipv6** keywords), unicast RPF is enabled for both IPv4 and IPv6. **Note** You can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. When you configure one version, all the mode changes must be done by this version. The interface blocks all the other versions.

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

Step 5	(Optional) **show ip interface ethernet** *slot*/*port* **Example:** `switch(config)# show ip interface ethernet 2/3`	Displays the IP information for an interface and verifies if the unicast RPF is enabled.
Step 6	(Optional) **show running-config interface ethernet** *slot*/*port* **Example:** `switch(config)# show running-config interface ethernet 2/3`	Displays the configuration for an interface in the running configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for Unicast RPF

The following examples show how to configure loose Unicast RPF for IPv4/IPv6 packets:

- ```
  interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
  ```
- ```
  interface Ethernet2/3
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via any
  ```

The following examples show how to configure strict Unicast RPF for IPv4/IPv6 packets:

- ```
  interface Ethernet2/2
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
  ```
- ```
  interface Ethernet2/2
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via rx
  ```

Verifying the Unicast RPF Configuration

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

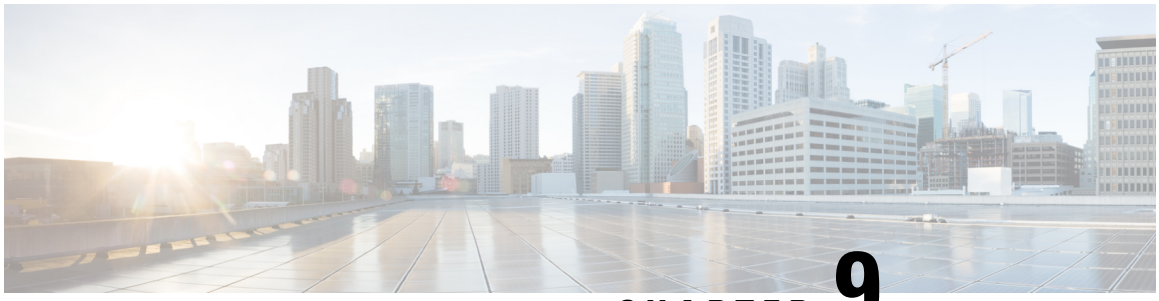
Command	Purpose
show running-config interface ethernet *slot/port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.
show running-config ip6 [**all**]	Displays the IPv6 configuration in the running configuration.
show startup-config interface ethernet *slot/port*	Displays the interface configuration in the startup configuration.
show ip interface ethernet *slot/port*	Displays the IP information for an interface and verifies if the unicast RPF is enabled or disabled.
show startup-config ip	Displays the IP configuration in the startup configuration.

Additional References for Unicast RPF

This section includes additional information related to implementing unicast RPF.

Related Documents

Related Topic	Document Title
MPLS VPN	Cisco Nexus 3600 Series NX-OS Label Switching Configuration Guide



CHAPTER 9

Configuring Control Plane Policing

This chapter contains the following sections:

About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic entering the switch from a non-management port. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, which can be perpetrated either inadvertently or maliciously, typically involve high rates of traffic destined to the supervisor module or CPU itself.

The supervisor module divides the traffic that it manages into three functional components or planes:

Data plane
: Handles all the data traffic. The basic functionality of a Cisco NX-OS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.

Control plane
: Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

Management plane
Runs the components meant for Cisco NX-OS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. For example, excessive traffic to the supervisor module could overload and slow down the performance of the entire Cisco NX-OS device. For example, a DoS attack on the supervisor module could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks include:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)
- High route processor or switch processor CPU utilization
- Route flaps due to loss of routing protocol updates or keepalives
- Unstable Layer 2 topology
- Slow or unresponsive interactive sessions with the CLI
- Processor resource exhaustion, such as the memory and buffers
- Indiscriminate drops of incoming packets

Caution It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

Control Plane Protection

To protect the control plane, the Cisco NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Cisco NX-OS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Control Plane Packet Types

Different types of packets can reach the control plane:

Receive packets
Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

Exception packets
Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

The following exceptions are possible from line cards only:

- match exception ip option
- match exception ipv6 option
- match exception ttl-failure

The following exceptions are possible from fabric modules only:

- match exception ipv6 icmp unreachable
- match exception ip icmp unreachable

The following exceptions are possible from line cards and fabric modules:

- match exception mtu-failure

Redirected packets
Packets that are redirected to the supervisor module.

Glean packets
If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco NX-OS device. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the supervisor module receives these packets.

Classification for CoPP

For effective protection, the Cisco NX-OS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set. You configure packet classifications and rate controlling policies using class maps and policy maps.

Rate Controlling Mechanisms

Once the packets are classified, the Cisco NX-OS device has different mechanisms to control the rate at which packets arrive at the supervisor module. Two mechanisms control the rate of traffic to the supervisor module. One is called policing and the other is called rate limiting.

Using hardware policers, you can define separate actions for traffic that conforms to or violates certain conditions. The actions can transmit the packet, mark down the packet, or drop the packet.

You can configure the following parameters for policing:

Committed information rate (CIR)
Desired bandwidth, specified as a bit rate or a percentage of the link rate.

Committed burst (BC)
Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling

In addition, you can set separate actions such as transmit or drop for conform and violate traffic.

For more information on policing parameters, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

Dynamic and Static CoPP ACLs

CoPP access control lists (ACLs) are classified as either dynamic or static. Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches use only dynamic CoPP ACLs. Cisco Nexus 9200 Series switches use both dynamic and static CoPP ACLs.

Dynamic CoPP ACLs work only for Forwarding Information Base (FIB)-based supervisor redirected packets, and static CoPP ACLs work for ACL-based supervisor redirected packets. Dynamic CoPP ACLs are supported for myIP and link-local multicast traffic, and static CoPP ACLs are supported for all other types of traffic.

Static CoPP ACLs are identified by a substring. Any ACL that has one of these substrings is categorized as a static CoPP ACL.

- MAC-based static CoPP ACL substrings:
  - acl-mac-cdp-udld-vtp
  - acl-mac-cfsoe
  - acl-mac-dot1x
  - acl-mac-l2-tunnel
  - acl-mac-l3-isis
  - acl-mac-lacp
  - acl-mac-lldp
  - acl-mac-sdp-srp
  - acl-mac-stp
  - acl-mac-undesirable
- Protocol-based static CoPP ACL substrings:
  - acl-dhcp
  - acl-dhcp-relay-response
  - acl-dhcp6
  - acl-dhcp6-relay-response
  - acl-ptp
- Multicast-based static CoPP ACL substrings:
  - acl-igmp

For more information on static CoPP ACLs, see Guidelines and Limitations for CoPP, on page 131 .

Default Policing Policies

When you bring up your Cisco NX-OS device for the first time, the Cisco NX-OS software installs the default copp-system-p-policy-strict policy to protect the supervisor module from DoS attacks. You can set the level of protection by choosing one of the following CoPP policy options from the initial setup utility:

- Strict—This policy is 1 rate and 2 color.
- Moderate—This policy is 1 rate and 2 color. The important class burst size is greater than the strict policy but less than the lenient policy.
- Lenient—This policy is 1 rate and 2 color. The important class burst size is greater than the moderate policy but less than the dense policy.
- Dense—This policy is 1 rate and 2 color. The policer CIR values are less than the strict policy.
- Skip—No control plane policy is applied. (Cisco does not recommend using the Skip option because it will impact the control plane of the network.)

If you do not select an option or choose not to execute the setup utility, the software applies strict policing. We recommend that you start with the strict policy and later modify the CoPP policies as required.

Note Strict policing is not applied by default when using POAP, so you must configure a CoPP policy.

The copp-system-p-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements. The default CoPP policy does not change when you upgrade the software.

Caution Selecting the skip option and not subsequently configuring CoPP protection can leave your Cisco NX-OS device vulnerable to DoS attacks.

You can reassign the CoPP default policy by entering the setup utility again using the **setup** command from the CLI prompt or by using the **copp profile** command.

Default Class Maps

The copp-system-class-critical class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-critical
      match access-group name copp-system-p-acl-bgp
      match access-group name copp-system-p-acl-rip
      match access-group name copp-system-p-acl-vpc
      match access-group name copp-system-p-acl-bgp6
      match access-group name copp-system-p-acl-ospf
      match access-group name copp-system-p-acl-rip6
      match access-group name copp-system-p-acl-eigrp
      match access-group name copp-system-p-acl-ospf6
      match access-group name copp-system-p-acl-eigrp6
      match access-group name copp-system-p-acl-auto-rp
      match access-group name copp-system-p-acl-mac-l3-isis
```

The copp-system-class-exception class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-exception
      match exception ip option
      match exception ip icmp unreachable
      match exception ipv6 option
      match exception ipv6 icmp unreachable
```

The copp-system-class-exception-diag class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-exception-diag
      match exception ttl-failure
      match exception mtu-failure
```

The copp-system-class-important class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-important
      match access-group name copp-system-p-acl-hsrp
      match access-group name copp-system-p-acl-vrrp
      match access-group name copp-system-p-acl-hsrp6
      match access-group name copp-system-p-acl-vrrp6
      match access-group name copp-system-p-acl-mac-lldp
```

The copp-system-class-l2-default class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l2-default
      match access-group name copp-system-p-acl-mac-undesirable
```

The copp-system-class-l2-unpoliced class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l2-unpoliced
      match access-group name copp-system-p-acl-mac-stp
      match access-group name copp-system-p-acl-mac-lacp
      match access-group name copp-system-p-acl-mac-cfsoe
      match access-group name copp-system-p-acl-mac-sdp-srp
      match access-group name copp-system-p-acl-mac-l2-tunnel
      match access-group name copp-system-p-acl-mac-cdp-udld-vtp
```

The copp-system-class-l3mc-data class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l3mc-data
      match exception multicast rpf-failure
      match exception multicast dest-miss
```

The copp-system-class-l3uc-data class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l3uc-data
      match exception glean
```

The copp-system-class-management class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-management
      match access-group name copp-system-p-acl-ftp
      match access-group name copp-system-p-acl-ntp
      match access-group name copp-system-p-acl-ssh
      match access-group name copp-system-p-acl-http
      match access-group name copp-system-p-acl-ntp6
      match access-group name copp-system-p-acl-sftp
      match access-group name copp-system-p-acl-snmp
      match access-group name copp-system-p-acl-ssh6
      match access-group name copp-system-p-acl-tftp
      match access-group name copp-system-p-acl-https
      match access-group name copp-system-p-acl-snmp6
      match access-group name copp-system-p-acl-tftp6
      match access-group name copp-system-p-acl-radius
      match access-group name copp-system-p-acl-tacacs
      match access-group name copp-system-p-acl-telnet
      match access-group name copp-system-p-acl-radius6
      match access-group name copp-system-p-acl-tacacs6
      match access-group name copp-system-p-acl-telnet6
```

The copp-system-class-monitoring class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-monitoring
      match access-group name copp-system-p-acl-icmp
      match access-group name copp-system-p-acl-icmp6
      match access-group name copp-system-p-acl-traceroute
```

The copp-system-class-multicast-host class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-multicast-host
      match access-group name copp-system-p-acl-mld
```

The copp-system-class-multicast-router class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-multicast-router
      match access-group name copp-system-p-acl-pim
      match access-group name copp-system-p-acl-msdp
      match access-group name copp-system-p-acl-pim6
      match access-group name copp-system-p-acl-pim-reg
      match access-group name copp-system-p-acl-pim6-reg
      match access-group name copp-system-p-acl-pim-mdt-join
```

The copp-system-class-nat-flow class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-nat-flow
      match exception nat-flow
```

The copp-system-class-ndp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-ndp
      match access-group name copp-system-p-acl-ndp
```

The copp-system-class-normal class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal
      match access-group name copp-system-p-acl-mac-dot1x
      match protocol arp
```

The copp-system-class-normal-dhcp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp
      match access-group name copp-system-p-acl-dhcp
      match access-group name copp-system-p-acl-dhcp6
```

The copp-system-class-normal-dhcp-relay-response class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp-relay-response
      match access-group name copp-system-p-acl-dhcp-relay-response
      match access-group name copp-system-p-acl-dhcp6-relay-response
```

The copp-system-class-normal-igmp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-igmp
      match access-group name copp-system-p-acl-igmp
```

The copp-system-class-redirect class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-redirect
      match access-group name copp-system-p-acl-ptp
```

The copp-system-class-undesirable class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-undesirable
      match access-group name copp-system-p-acl-undesirable
      match exception multicast sg-rpf-failure
```

The copp-system-class-fcoe class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-fcoe
      match access-group name copp-system-p-acl-mac-fcoe
```

Note The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.

Strict Default CoPP Policy

On Cisco Nexus 9200 Series switches, the strict CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
```

```
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the strict CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
```

```
    set cos 1
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 32 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 32 packets conform transmit violate drop
```

Moderate Default CoPP Policy

On Cisco Nexus 9200 Series switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
```

```
    set cos 1
    police cir 1300 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 96000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
```

```
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 96 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
```

Lenient Default CoPP Policy)

On Cisco Nexus 9200 Series switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 64000 bytes conform transmit violate drop
```

```
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
```

```
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
```

Dense Default CoPP Policy

On Cisco Nexus 9200 Series switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
```

```
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 4500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 370 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 2500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 300 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 600 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 350 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 750 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 750 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 1400 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 100 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
```

```
  set cos 1
  police cir 250 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-critical
  set cos 7
  police cir 2500 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-important
  set cos 6
  police cir 1200 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-multicast-router
  set cos 6
  police cir 1200 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-management
  set cos 2
  police cir 1200 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-multicast-host
  set cos 2
  police cir 1000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l3mc-data
  set cos 1
  police cir 1200 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal
  set cos 1
  police cir 750 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-ndp
  set cos 1
  police cir 750 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp
  set cos 1
  police cir 150 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 200 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 2500 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 50 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 750 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
class class-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
```

Packets Per Second Credit Limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

Modular QoS Command-Line Interface

CoPP uses the Modular Quality of Service Command-Line Interface (MQC). MQC is a CLI structure that allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the CoPP feature that will be applied to the traffic class.

Procedure

Step 1 Define a traffic class using the **class-map** command. A traffic class is used to classify traffic.

This example shows how to create a new class-map called copp-sample-class:

```
class-map type control-plane copp-sample-class
```

Step 2 Create a traffic policy using the **policy-map** command. A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

This example shows how to attach the policy map to the control plane:

```
control-plane
service-policy input copp-system-policy
```

Note The copp-system-policy is always configured and applied. There is no need to use this command explicitly.

CoPP and the Management Interface

The Cisco NX-OS device supports only hardware-based CoPP, which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- Before you upgrade to Release 9.3(7), set the scale factor to 1 using the **scale-factor 1 module** *multiple-module-range* command.

- We recommend that you use the strict default CoPP policy initially and then later modify the CoPP policies based on the data center and application requirements.
- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features used in your specific environment as well as the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.
- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.
- All the traffic that you do not specify in the other class maps is put into the last class, the default class. Monitor the drops in this class and investigate if these drops are based on traffic that you do not want or the result of a feature that was not configured and you need to add.
- All broadcast traffic is sent through CoPP logic in order to determine which packets (for example, ARP and DHCP) need to be redirected through an access control list (ACL) to the router processor. Broadcast traffic that does not need to be redirected is matched against the CoPP logic, and both conforming and violated packets are counted in the hardware but not sent to the CPU. Broadcast traffic that needs to be sent to the CPU and broadcast traffic that does not need to be sent to the CPU must be separated into different classes.
- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the device. Filtering this traffic could prevent remote access to the Cisco NX-OS device and require a console connection.
- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (you cannot use the **service-policy output copp** command to the control plane interface).
- You can use the access control entry (ACE) hit counters in the hardware only for ACL logic. Use the software ACE hit counters and the **show access-lists** and **show policy-map type control-plane** commands to evaluate CPU traffic.
- The Cisco NX-OS device hardware performs CoPP on a per-forwarding-engine basis. CoPP does not support distributed policing. Therefore, you should choose rates so that the aggregate traffic does not overwhelm the supervisor module.
- If multiple flows map to the same class, individual flow statistics will not be available.
- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with additional classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available.
- Before you downgrade from a Cisco NX-OS release that supports the CoPP feature to an earlier Cisco NX-OS release that supports the CoPP feature, you should verify compatibility using the **show incompatibility nxos bootflash:***filename* command. If an incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.
- You cannot disable CoPP. If you attempt to disable it, packets are rate limited at 50 packets per seconds [for releases prior to Cisco NX-OS Release 7.0(3)I2(1)], or an error message appears [starting with Cisco NX-OS Release 7.0(3)I2(1)].

- Cisco Nexus 9200 Series switches support CoPP policer rates only in multiples of 10 kbps. If a rate is configured that is not a multiple of 10 kbps, the rate is rounded down. For example, the switch will use 50 kbps if a rate of 55 kbps is configured. (The **show policy-map type control-plane** command shows the user configured rate. See Verifying the CoPP Configuration, on page 141 for more information.)
- For Cisco Nexus 9200 Series switches, ip icmp redirect, ipv6 icmp redirect, ip icmp unreachable, ipv6 icmp unreachable, and mtu-failure use the same TCAM entry, and they will all be classified to the class map where the first exception is present in the policy. In the CoPP strict profile, they are classified to the class-exception class map. In a different CoPP policy, if the first exception is in a different class map (for example, class-exception-diag), the rest of the exceptions will be classified to the same class map.
- The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.
- The following guidelines and limitations apply to static CoPP ACLs:
  - Only Cisco Nexus 9200 Series switches use static CoPP ACLs.
  - Static CoPP ACLs can be remapped to a different CoPP class.
  - Access control entries (ACEs) cannot be modified or removed for static CoPP ACLs.
  - If a CoPP ACL has a static ACL substring, it will be mapped to that type of traffic. For example, if the ACL includes the acl-mac-stp substring, STP traffic will be classified to the class map for that ACL.
  - Static CoPP ACLs take priority over dynamic CoPP ACLs, regardless of their position in the CoPP policy, the order in which they are configured, and how they appear in the output of the **show policy-map type control-plane** command.
  - You must have static CoPP ACLs in the CoPP policy. Otherwise, the CoPP policy will be rejected.

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

Default Settings for CoPP

This table lists the default settings for CoPP parameters.

Table 15: Default CoPP Parameters Settings

Parameters	Default
Default policy	Strict
Default policy	9 policy entries **Note** The maximum number of supported policies with associated class maps is 128.
Scale factor value	1.00

Configuring CoPP

This section describes how to configure CoPP.

Configuring a Control Plane Class Map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

You can configure policies for IP version 4 (IPv4) and IP version 6 (IPv6) packets.

Before you begin

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

Procedure

	Command or Action	Purpose

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

Step 2	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
	Example:
	`switch(config)# class-map type control-plane ClassMapA`
	`switch(config-cmap)#`
		Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.
		Note You cannot use class-default, match-all, or match-any as class map names.

Step 3	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name MyAccessList`	Specifies matching for an IP ACL. **Note** The permit and deny ACL keywords are ignored in the CoPP matching.
Step 4	(Optional) **match exception** {**ip** | **ipv6**} **icmp redirect** **Example:** `switch(config-cmap)# match exception ip icmp redirect`	Specifies matching for IPv4 or IPv6 ICMP redirect exception packets.

	Command or Action	Purpose
Step 5	(Optional) **match exception** {**ip** | **ipv6**} **icmp unreachable** **Example:** `switch(config-cmap)# match exception ip icmp unreachable`	Specifies matching for IPv4 or IPv6 ICMP unreachable exception packets.
Step 6	(Optional) **match exception** {**ip** | **ipv6**} **option** **Example:** `switch(config-cmap)# match exception ip option`	Specifies matching for IPv4 or IPv6 option exception packets.
Step 7	**match protocol arp** **Example:** `switch(config-cmap)# match protocol arp`	Specifies matching for IP Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) packets.

Step 8	**exit**
	Example:
	`switch(config-cmap)# exit`
	`switch(config)#`
		Exits class map configuration mode.

Step 9	(Optional) **show class-map type control-plane** [*class-map-name*] **Example:** `switch(config)# show class-map type control-plane`	Displays the control plane class map configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the following default is configured:

- 50 packets per second (pps) with a burst of 32 packets (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches)
- 150 kilobits per second (kbps) with a burst of 32,000 bytes (for Cisco Nexus 9200 Series switches)

Before you begin

Ensure that you have configured a control plane class map.

Procedure

	Command or Action	Purpose

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

Step 2	**policy-map type control-plane** *policy-map-name*
	Example:
	`switch(config)# policy-map type control-plane ClassMapA`
	`switch(config-pmap)#`
		Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive.

Step 3	**class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
	Example:
	`switch(config-pmap)# class ClassMapA`
	`switch(config-pmap-c)#`
		Specifies a control plane class map name or the class default and enters control plane class configuration mode.
		The class-default class map is always at the end of the class map list for a policy map.

Step 4	Enter one of the following commands:
	• **police** [**cir**] {*cir-rate* [*rate-type*]}
	• **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
	• **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]
	Example:
	`switch(config-pmap-c)# police cir 52000 bc 1000 packets`
	Example:
	`switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes`
		Specifies the committed information rate (CIR). The rate range is as follows:
		• 0 to 268435456 pps (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches)
		• 0 to 80000000000 bps/gbps/kbps/mbps (for Cisco Nexus 9200 Series switches)
		Note The CIR rate range starts with 0. In previous releases, the CIR rate range starts with 1. A value of 0 drops the packet.
		The committed burst (BC) range is as follows:
		• 1 to 1073741 packets (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches)
		• 1 to 512000000 bytes/kbytes/mbytes (for Cisco Nexus 9200 Series switches)
		The **conform transmit** action transmits the packet.
		Note You can specify the BC and conform action for the same CIR.

	Command or Action	Purpose
Step 5	(Optional) **logging drop threshold** [*drop-count* [**level** *syslog-level*]] **Example:** `switch(config-pmap-c)# logging drop threshold 100`	Specifies the threshold value for dropped packets and generates a syslog if the drop count exceeds the configured threshold. The range for the *drop-count* argument is from 1 to 8000000000 bytes. The range for the *syslog-level* argument is from 1 to 7, and the default level is 4.
Step 6	(Optional) **set cos** *cos-value* **Example:** `switch(config-pmap-c)# set cos 1`	Specifies the 802.1Q class of service (CoS) value. The range is from 0 to 7. The default value is 0.

Step 7	**exit**
	Example:
	`switch(config-pmap-c)# exit`
	`switch(config-pmap)#`
		Exits policy map class configuration mode.

Step 8	**exit**
	Example:
	`switch(config-pmap)# exit`
	`switch(config)#`
		Exits policy map configuration mode.

Step 9	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:** `switch(config)# show policy-map type control-plane`	Displays the control plane policy map configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Control Plane Service Policy

You can configure one or more policy maps for the CoPP service policy.

Note When you try to change the CoPP policy and apply a custom CoPP policy, it is configured in the hardware as non-atomic, and the following system message appears:

```
This operation can cause disruption of control traffic. Proceed (y/n)?  [no] y
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT24-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT23-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT21-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT25-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT26-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT22-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT4-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
```

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

	Command or Action	Purpose

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

Step 2	**control-plane**
	Example:
	`switch(config)# control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	[**no**] **service-policy input** *policy-map-name* **Example:** `switch(config-cp)# service-policy input PolicyMapA`	Specifies a policy map for the input traffic. Repeat this step if you have more than one policy map. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds.

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

	Command or Action	Purpose
Step 5	(Optional) **show running-config copp** [**all**] **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the CoPP Scale Factor Per Line Card

You can configure the CoPP scale factor per line card.

The scale factor configuration is used to scale the policer rate of the applied CoPP policy for a particular line card. The accepted value is from 0.10 to 2.00. You can increase or reduce the policer rate for a particular line card without changing the current CoPP policy. The changes are effective immediately, so you do not need to reapply the CoPP policy.

Procedure

	Command or Action	Purpose

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

Step 2	**control-plane**
	Example:
	`switch(config)# control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	**scale-factor** *value* **module** *multiple-module-range* **Example:** `switch(config-cp)# scale-factor 1.10 module 1-2`	Configures the policer rate per line card. The allowed scale factor value is from 0.10 to 2.00. When the scale factor value is configured, the policing values are multiplied by the corresponding scale factor value of the module, and it is programmed in the particular module. To revert to the default scale factor value of 1.00, use the **no scale-factor** *value* **module** *multiple-module-range* command, or explicitly set the default scale factor value to 1.00 using the **scale-factor 1 module** *multiple-module-range* command.

	Command or Action	Purpose
Step 4	(Optional) **show policy-map interface control-plane** **Example:** `switch(config-cp)# show policy-map interface control-plane`	Displays the applied scale factor values when a CoPP policy is applied.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing or Reapplying the Default CoPP Policy

You can change to a different default CoPP policy, or you can reapply the same default CoPP policy.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **copp profile** [**strict** | **moderate** | **lenient** | **dense**] **Example:** `switch(config)# copp profile moderate`	Applies the CoPP best practice policy. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds.
Step 2	(Optional) **show copp status** **Example:** `switch(config)# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the CoPP best practice policy is attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration in the running configuration.

Copying the CoPP Best Practice Policy

The CoPP best practice policy is read-only. If you want to modify its configuration, you must copy it.

Procedure

	Command or Action	Purpose
Step 1	**copp copy profile** {**strict** | **moderate** | **lenient** | **dense**} {**prefix** | **suffix**} *string* **Example:**	Creates a copy of the CoPP best practice policy. CoPP renames all class maps and policy maps with the specified prefix or suffix.

	Command or Action	Purpose
	`switch# copp copy profile strict prefix abc`	
Step 2	(Optional) **show copp status** **Example:** `switch# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the copied policy is not attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch# show running-config copp`	Displays the CoPP configuration in the running configuration, including the copied policy configuration.

Verifying the CoPP Configuration

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

Command	Purpose
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps and CIR and BC values.
show policy-map interface control-plane	Displays the policy values with associated class maps and drops per policy or class map. It also displays the scale factor values when a CoPP policy is applied. When the scale factor value is the default (1.00), it is not displayed. **Note** The scale factor changes the CIR and BC values internally on each module, but the display shows the configured CIR and BC values only. The actual applied value on a module is the scale factor multiplied by the configured value.
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.

Command	Purpose
show copp diff profile {**strict** | **moderate** | **lenient** | **dense**} [**prior-ver**] **profile** {**strict** | **moderate** | **lenient** | **dense**} **show copp diff profile**	Displays the difference between two CoPP best practice policies. When you do not include the prior-ver option, this command displays the difference between two currently applied default CoPP best practice policies (such as the currently applied strict and currently applied moderate policies). When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies).
show copp profile {**strict** | **moderate** | **lenient** | **dense**}	Displays the details of the CoPP best practice policy, along with the classes and policer values.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Displaying the CoPP Configuration Status

Procedure

	Command or Action	Purpose
Step 1	switch# **show copp status**	Displays the configuration status for the CoPP feature.

Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

Monitoring CoPP

Procedure

	Command or Action	Purpose
Step 1	switch# **show policy-map interface control-plane**	Displays packet-level statistics for all classes that are part of the applied CoPP policy. Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting).

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

  Service-policy  input: copp-system-p-policy-strict

    class-map copp-system-p-class-critical (match-any)
      set cos 7
      police cir 19000 pps , bc 128 packets
      module 4 :
        transmitted 373977 packets;
        dropped 0 packets;
```

Clearing the CoPP Statistics

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

Configuration Examples for CoPP

This section includes example CoPP configurations.

CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-p-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-p-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-p-acl-arp
permit any any 0x0806

ip access-list copp-system-p-acl-tacas
permit udp any any eq 49

ip access-list copp-system-p-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-p-acl-icmp
permit icmp any any

class-map type control-plane match-any copp-system-p-class-critical
match access-group name copp-system-p-acl-igmp
match access-group name copp-system-p-acl-msdp

class-map type control-plane match-any copp-system-p-class-normal
match access-group name copp-system-p-acl-icmp
match exception ip icmp redirect
```

```
match exception ip icmp unreachable
match exception ip option

policy-map type control-plane copp-system-p-policy

class copp-system-p-class-critical
police cir 19000 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-important
police cir 500 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-normal
police cir 300 pps bc 32 packets conform transmit violate drop

class class-default
police cir 50 pps bc 32 packets conform transmit violate drop

control-plane
service-policy input copp-system-p-policy
```

Create CoPP class and associate ACL:

```
class-map type control-plane copp-arp-class
match access-group name copp-arp-acl
```

Add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
class copp-arp-class
police pps 500
```

Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup

         ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.


*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.


Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no)[y]: <CR>

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n
```

```
  Enter the switch name : <CR>

  Enable license grace period? (yes/no) [n]: n

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway? (yes/no) [y]: n

  Configure advanced IP options? (yes/no) [n]: <CR>

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: <CR>

    Type of ssh key you would like to generate (dsa/rsa) : <CR>

  Configure the ntp server? (yes/no) [n]: n

  Configure default interface layer (L3/L2) [L3]: <CR>

  Configure default switchport interface state (shut/noshut) [shut]: <CR>

  Configure best practices CoPP profile (strict/moderate/lenient/dense/skip) [strict]:
strict


The following configuration will be applied:
  password strength-check
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-p-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

Additional References for CoPP

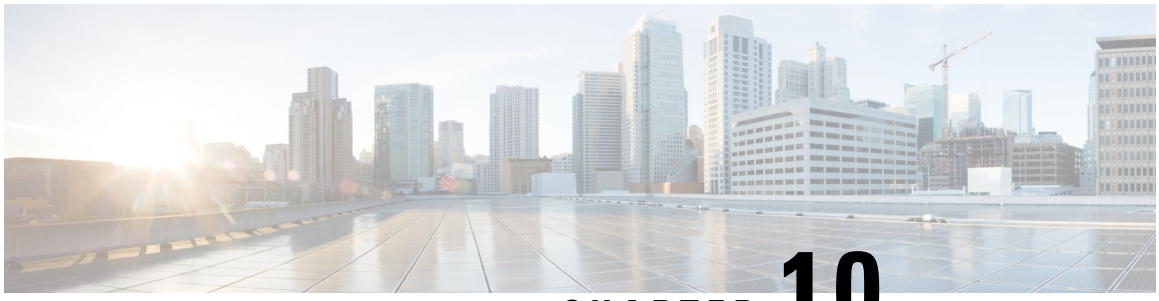
This section provides additional information related to implementing CoPP.

Related Documents

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*

Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker



CHAPTER 10

Configuring MACsec

This document describes how to configure MACsec on Cisco NX-OS devices.

Configuring MACsec

This document describes how to configure MACsec on Cisco NX-OS devices.

About MACsec

Media Access Control Security (MACsec) an IEEE 802.1AE along with MACsec Key Agreement (MKA) protocol provide secure communications on Ethernet links. It offers the following :

- Provides line rate encryption capabilities.
- Helps to ensure data confidentiality by providing strong encryption at Layer 2.
- Provides integrity checking to help ensure that data cannot be modified in transit.
- Can be selectively enabled using a centralized policy to help ensure that it is enforced where required while allowing non-MACsec-capable components to access the network.
- Encrypts packets on a hop-by-hop basis at Layer 2, allowing the network to inspect, monitor, mark, and forward traffic according to your existing policies, unlike end-to-end Layer 3 encryption techniques that hide the contents of packets from the network devices they cross.

Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the keychain after the lifetime expires. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Configuring a MACsec Keychain and Keys, on page 151.

A key can roll over to a second key within the same keychain by configuring the second key and a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the

list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless, that is, the key rolls over without traffic interruption.

Fallback Key

A MACsec session can fail due to a key/key name (CKN) mismatch or a finite key duration between the switch and a peer. If a MACsec session does fail, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

To configure a MACsec fallback key, see Configuring MACsec Fallback Key, on page 153.

Guidelines and Limitations for MACsec

MACsec has the following guidelines and limitations:

- MACsec is supported on the following interface types:
  - Layer 2 switch ports (access and trunk)
  - Layer 3 routed interfaces (no subinterfaces)

Note Enabling MACsec on the Layer 3 routed interface also enables encryption on all the subinterfaces that are defined under that interface. However, selectively enabling MACsec on a subset of subinterfaces of the same Layer 3 routed interface is not supported.

  - Individual Layer 2 and Layer 3-port channel members (no subinterfaces)
- Secure Channel Identified (SCI) encoding cannot be disabled on Cisco Nexus 3600 Series switches.
- Support for MACsec is not available for Cisco Nexus ToR switches when you downgrade from Release 10.x.
- MKA is the only supported key exchange protocol for MACsec. The Security Association Protocol (SAP) is not supported.
- Link-level flow control (LLFC) and priority flow control (PFC) are not supported with MACsec.
- Multiple MACsec peers (different SCI values) for the same interface are not supported.
- You can retain the MACsec configuration when you disable MACsec using the **macsec shutdown** command.
- MACsec sessions are liberal in accepting packets from a key server whose latest Rx and latest Tx flags have been retired after Tx SA installation for the first time. The MACsec session then converges into a secure state.
- Beginning with Cisco NX-OS Release 10.1(1), you can modify MACSec policy while the policy is referenced by an interface.
- Beginning with Cisco Nexus Release 10.1(1), MACsec is supported on the Cisco Nexus N3KC3636C-R platform switches.

- N3K-C3636C-R—MACsec is supported on the following eight ports of N3K-C3636C-R, marked in green [Ports 29–36].

Note On the Cisco N3K-C3636C-R platform switches, when MACsec is either configured or unconfigured on a port, there will be a port-flap occurrence irrespective of MACsec security-policy type.

- Cisco Nexus 3600 Series switches do not support MACsec on any of the MACsec capable ports when QSA is being used.
- MACsec is not supported on breakout ports, and breakout is not supported on the following eight ports, from Port 29 to Port 36, of N3K-C3636C-R when MACsec is configured.
- Packet drops for a short period when the conf-offset parameter is changed dynamically for a MACsec policy. Change the conf-offset parameter only in static configuration when the policy is not active on the port.
- MACsec is not supported on Cisco Nexus N3K-C36180YC-R platform switches.

Keychain Restrictions:

- You cannot overwrite the octet string for a MACsec key. Instead, you must create a new key or a new keychain.
- A new key in the keychain is configured when you enter end or exit. The default timeout for editor mode is 6 seconds. If the key is not configured with the key octet string or/and the send lifetime within the 6-second window, incomplete information may be used to bring up the MACsec session and could result in the session being stuck in an Authorization Pending state. If the MACsec sessions are not converged after the configuration is complete, you might be advised to shut/no shut the ports.
- For a given keychain, key activation times should overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.

Fallback Restrictions:

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and will show as rekeying on the old CA under status. And the MACsec session on the new key on primary PSK will be in init state.
- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.
- The key ID (CKN) used in the fallback key chain must not match any of the key IDs (CKNs) used in the primary key chain.
- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec Policy Restrictions:

- BPDU packets can be transmitted before a MACsec session becomes secure.

Layer 2 Tunneling Protocol (L2TP) Restrictions:

- MACsec is not supported on ports configured for dot1q tunneling or L2TP.
- L2TP does not work if STP is enabled on trunk ports for non-native VLANs.

Statistics Restrictions:

- Few CRC errors should occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).
- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.

Enabling MACsec

Before you can access the MACsec and MKA commands, you must enable the MACsec feature.

Procedure

	Command or Action	Purpose

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

Step 2	**feature macsec** **Example:** `switch(config)# feature macsec`	Enables MACsec and MKA on the device.
Step 3	(Optional) copy running-config startup-config **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling MACsec

Beginning with Cisco NX-OS Release 10.1(1), disabling the MACsec feature only deactivates this feature and does not remove the associated MACsec configurations.

Disabling MACsec has the following conditions:

- MACsec shutdown is global command and is not available at the interface level.

- The macsec shutdown, show macsec mka session/summary, show macsec mka session detail, and show macsec mka/secy statistics commands will display the 'Macsec is shutdown' message. However, the show macsec policy and show key chain commands will display the output.
- Consecutive MACsec status changes from macsec shutdown to no macsec shutdown and vice versa needs a 30 seconds time interval in between the status change.

Procedure

	Command or Action	Purpose

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

Step 2	**macsec shutdown** **Example:** `switch(config)# macsec shutdown`	Disables the MACsec configuration on the device. The **no** option restores the MACsec feature.
Step 3	(Optional) copy running-config startup-config **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. This step is required only if you want to retain the MACsec in the shutdown state after the switch reload. **Note** You can also disable the MACsec feature using the **no feature macsec** command.

Configuring a MACsec Keychain and Keys

You can create a MACsec keychain and keys on the device.

Note Only MACsec keychains will result in converged MKA sessions.

Before you begin

Make sure that MACsec is enabled.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	(Optional) [**no**] **key-chain macsec-psk no-show** **Example:** `switch(config)# key-chain macsec-psk no-show`	Hides the encrypted key octet string in the output of the **show running-config** and **show startup-config** by replacing the string with a wildcard character. By default, PSK keys are displayed in encrypted format and can be easily decrypted. This command applies only to MACsec keychains. **Note** The octet string is also hidden when you save the configuration to a file.

Step 3	**key chain** *name* **macsec**
	Example:
	`switch(config)# key chain 1 macsec`
	`switch(config-macseckeychain)#`
		Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.

Step 4	**key** *key-id*
	Example:
	`switch(config-macseckeychain)# key 1000`
	`switch(config-macseckeychain-macseckey)#`
		Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64.
		Note The key must consist of an even number of characters.

Step 5	**key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**} **Example:** `switch(config-macseckeychain-macseckey)# key-octet-string abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm AES_256_CMAC`	Configures the octet string for the key. The octet-string argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command. The key octet string includes the following: • 0 Encryption Type - No encryption (default) • 6 Encryption Type - Proprietary (Type-6 encrypted). For more information, see *Enabling Type-6 Encryption on MACsec Keys*. • 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters

	Command or Action	Purpose
		Note MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. To interoperate between previous releases and Cisco NX-OS Release 7.0(3)I7(2) or a later release, you must use keys with the AES_256_CMAC cryptographic algorithm.
Step 6	**send-lifetime** *start-time* **duration** *duration* **Example:** `switch(config-macseckeychain-macseckey)# send-lifetime 00:00:00 Oct 04 2016 duration 100000`	Configures a send lifetime for the key. By default, the device treats the start time as UTC. The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
Step 7	(Optional) **show key chain** *name* **Example:** `switch(config-macseckeychain-macseckey)# show key chain 1`	Displays the keychain configuration.
Step 8	(Optional) copy running-config startup-config **Example:** `switch(config-macseckeychain-macseckey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring MACsec Fallback Key

Beginning with Cisco NX-OS Release 10.1(1), you can configure a fallback key on the device to initiate a backup session if the primary session fails as a result of a key/key name (CKN) mismatch or a finite key duration between the switch and peer.

Before you begin

Make sure that MACsec is enabled and a primary and fallback keychain and key ID are configured. See Configuring a MACsec Keychain and Keys, on page 151.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface** *name*
	Example:
	`switch(config)# interface ethernet 1/29`
	`switch(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.

Step 3	**macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name* **Example:** `switch(config-if)# macsec keychain kc2 policy abc fallback-keychain fb_kc2`	Specifies the fallback keychain to use after a MACsec session failure due to a key/key ID mismatch or a key expiration. The fallback key ID should not match any key ID from a primary keychain. Fallback keychain configuration for each interface can be changed on the corresponding interface, without removing the MACsec configuration, by reissuing the same command with the fallback keychain name changed. **Note** The command must be entered exactly the same as the existing configuration command for the interface, except for the fallback keychain name. See Configuring a MACsec Keychain and Keys, on page 151.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Make sure that MACsec is enabled.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**macsec policy** *name*
	Example:
	`switch(config)# macsec policy abc`
	`switch(config-macsec-policy)#`
		Creates a MACsec policy.

Step 3	**cipher-suite** *name* **Example:** `switch(config-macsec-policy)# cipher-suite GCM-AES-256`	Configures one of the following ciphers: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256.
Step 4	**key-server-priority** *number* **Example:** `switch(config-macsec-policy)# key-server-priority 0`	Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.
Step 5	**security-policy** *name* **Example:** `switch(config-macsec-policy)# security-policy should-secure`	Configures one of the following security policies to define the handling of data and control packets: • **must-secure**—Packets not carrying MACsec headers will be dropped. • **should-secure**—Packets not carrying MACsec headers will be permitted. This is the default value.
Step 6	**window-size** *number* **Example:** `switch(config-macsec-policy)# window-size 512`	Configures the replay protection window such that the secured interface will not accept any packet that is less than the configured window size. The range is from 0 to 596000000.
Step 7	**sak-expiry-time** *time* **Example:** `switch(config-macsec-policy)# sak-expiry-time 100`	Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0.
Step 8	**conf-offset** *name* **Example:** `switch(config-macsec-policy)# conf-offset CONF-OFFSET-0`	Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50. This command might be necessary for intermediate switches to use packet headers {dmac, smac, etype} like MPLS tags.
Step 9	(Optional) **show macsec policy** **Example:** `switch(config-macsec-policy)# show macsec policy`	Displays the MACsec policy configuration.

	Command or Action	Purpose
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-macsec-policy)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Rotating PSKs

Follow this procedure to rotate PSKs when the SAK expiry time is configured for 60 seconds in the MACsec policy.

Procedure

Step 1 Use the **no sak-expiry-time** command to remove the SAK expiry timer from the MACsec policy.

You need to remove the SAK expiry timer only for the number of policies in the configuration. You do not need to remove it for each interface. If you have defined only one policy and applied it to all interfaces, you need to remove the SAK expiry timer only from this policy.

Step 2 Wait for 2 minutes.

Step 3 Use the **key** *key-id* command to program the new key under the keychain.

Step 4 Once the session with the new key is secured, use the **no key** *key-id* command to delete the old key.

Step 5 Wait for 2 minutes.

Step 6 Use the **sak-expiry-timer 60** command to add the SAK rekey timer to the MACsec policy.

Verifying the MACsec Configuration

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

Command	Purpose
show key chain *name*	Displays the keychain configuration.
show macsec mka session [**interface** *type slot/port*] [**detail**]	Displays information about the MACsec MKA session for a specific interface or for all interfaces.
show macsec mka session details	Displays information about the MAC address.
show macsec mka summary	Displays the MACsec MKA configuration.
show macsec policy [*policy-name*]	Displays the configuration for a specific MACsec policy or for all MACsec policies.
show running-config macsec	Displays the running configuration information for MACsec.

The following example displays information about the MACsec MKA session for all interfaces.

```
switch(config)# show macsec mka session
Interface          Local-TxSCI                     # Peers            Status
Key-Server         Auth Mode
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/29       6c8b.d3db.e968/0001             1                  Secured
No                 PRIMARY-PSK
Ethernet1/30       6c8b.d3db.e96c/0001             1                  Secured
No                 PRIMARY-PSK
Ethernet1/31       6c8b.d3db.e970/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet1/32       6c8b.d3db.e974/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet1/33       6c8b.d3db.e978/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet1/34       6c8b.d3db.e97c/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet1/35       6c8b.d3db.e980/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet1/36       6c8b.d3db.e984/0001             1                  Secured
No                 PRIMARY-PSK
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 8
        Secured Sessions : 8
        Pending Sessions : 0
switch(config)#
```

The following example displays information about the MACsec MKA session for a specific interface. In addition to the common elements of the table as described in the previous example, the following also identifies the authentication mode which defines the current MACsec session type.

```
switch(config)# show macsec mka session interface e1/35
Interface          Local-TxSCI                     # Peers            Status
Key-Server         Auth Mode
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/35       6c8b.d3db.e980/0001             1                  Secured
Yes                PRIMARY-PSK
switch(config)#
```

The following example displays detailed information about the MACsec MKA session for a specific Ethernet interface:

```
switch(config)# show macsec mka session interface e1/35 details
Detailed Status for MKA Session
-------------------------------
Interface Name                         : Ethernet1/35
    Session Status                     : SECURED - Secured MKA Session with MACsec
    Local Tx-SCI                       : 6c8b.d3db.e980/0001
    Local Tx-SSCI                      : 2
    MKA Port Identifier                : 2
    CAK Name (CKN)                     : 2006
    CA Authentication Mode             : PRIMARY-PSK
    Member Identifier (MI)             : 50BE8367F1C6D0AB1C442229
    Message Number (MN)                : 1048
    MKA Policy Name                    : mpsr1
    Key Server Priority                : 1
    Key Server                         : Yes
    Include ICV                        : Yes
    SAK Cipher Suite                   : GCM-AES-128
    SAK Cipher Suite (Operational)     : GCM-AES-128
    Replay Window Size                 : 148809600
```

```
    Confidentiality Offset              : CONF-OFFSET-30
    Confidentiality Offset (Operational): CONF-OFFSET-30
    Latest SAK Status                   : Rx & TX
    Latest SAK AN                       : 0
    Latest SAK KI                       : 50BE8367F1C6D0AB1C44222900000021
    Latest SAK KN                       : 33
    Last SAK key time                   : 11:23:53 pst Tue Dec 15 2020
    CA Peer Count                       : 1
    Eapol dest mac                      : 0180.c200.0003
    Ether-type                          : 0x888e
Peer Status:
    Peer MI                             : 37AFE73EC8617FD32F70E21A
    RxSCI                               : 6c8b.d3db.e984/0001
    Peer CAK                            : Match
    Latest Rx MKPDU                     : 11:24:52 pst Tue Dec 15 2020
Fallback Data:
    Fallback CKN                        : FB2004
    Fallback MI                         : 849D72D5F6A900F5B0718C78
    Fallback MN                         : 0x3d6
Fallback Peer:
    Peer MI                             : 8DCE8CBE67B474D2C2955F58
    RxSCI                               : 6c8b.d3db.e984/0001
    Peer CAK                            : Match
    Latest Rx MKPDU                     : 11:24:52 pst Tue Dec 15 2020
switch(config)#
```

The following example displays the MACsec MKA configuration:

```
switch# show macsec mka summary
Interface          MACSEC-policy                   Keychain
------------------ ------------------------------- ------------------------
Ethernet2/13       1                                1/10000000000000000
Ethernet2/14       1                                1/10000000000000000
switch#
```

The following example displays the configuration for all MACsec policies:

```
switch# show macsec policy
MACSec Policy Cipher Pri Window Offset Security SAK Rekey time ICV Indicator
------------------------------- ---------------- ---- ------------ -------- --------------
-------------- -------------
system-default-macsec-policy GCM-AES-XPN-256 16 148809600 0 should-secure
pn-rollover FALSE
tests1 GCM-AES-XPN-256 16 148809600 0 should-secure
pn-rollover FALSE
tests2 GCM-AES-XPN-256 16 148809600 0 should-secure
pn-rollover FALSE
tests3 GCM-AES-256 16 148809600 0 should-secure
pn-rollover FALSE
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is not configured:

```
key chain KC256-1 macsec
    key 2000
      key-octet-string 7
075e701e1c5a4a5143475e5a527d7c7c706a6c724306170103555a5c57510b051e47080
a05000101005e0e50510f005c4b5f5d0b5b070e234e4d0a1d0112175b5e cryptographic-algorithm
AES_256_CMAC
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is configured:

```
key chain KC256-1 macsec
    key 2000
      key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

Displaying MACsec Statistics

You can display MACsec statistics using the following commands.

Command	Purpose
show macsec mka statistics [**interface** *type slot/port*]	Displays MACsec MKA statistics.
show macsec secy statistics [**interface** *type slot/port*]	Displays MACsec security statistics.

The following example shows the MACsec MKA statistics for a specific Ethernet interface:

```
switch# show macsec mka statistics interface ethernet 1/29
MKA Statistics for Session on interface (Ethernet1/29)
=======================================================
CA Statistics
   Pairwise CAK Rekeys..... 0

SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 41
      "Distributed SAK".. 0
   MKPDUs Validated & Rx... 41
      "Distributed SAK".. 0

MKA IDB Statistics
   MKPDUs Tx Success.......... 82
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUS No Tx on intf down.. 0
   MKPDUS No Rx on intf down.. 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 82

MKPDU Failures
   MKPDU Rx Validation ..................... 0
   MKPDU Rx Bad Peer MN..................... 0
   MKPDU Rx Non-recent Peerlist MN.......... 0
   MKPDU Rx Drop SAKUSE, KN mismatch........ 0
   MKPDU Rx Drop SAKUSE, Rx Not Set......... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.... 0
   MKPDU Rx Drop SAKUSE, AN Not in Use...... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set... 0
   MKPDU Rx Drop Packet, Ethertype Mismatch. 0
   MKPDU Rx Drop Packet, DestMAC Mismatch... 0

SAK Failures
   SAK Generation................... 0
   Hash Key Generation.............. 0
   SAK Encryption/Wrap.............. 0
   SAK Decryption/Unwrap............ 0

CA Failures
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0
```

```
MACsec Failures
   Rx SA Installation.............. 0
   Tx SA Installation.............. 0

switch(config)#
```

The following example shows the MACsec security statistics for a specific Ethernet interface.

Note The following differences exist for uncontrolled and controlled packets in Rx and Tx statistics:

- Rx statistics:
  - Uncontrolled = Encrypted and unencrypted
  - Controlled = Decrypted
- Tx statistics:
  - Uncontrolled = Unencrypted
  - Controlled = Encrypted
  - Common = Encrypted and unencrypted

```
switch(config)# show macsec secy statistics interface e1/29
Interface Ethernet1/29 MACSEC SecY Statistics:
------------------------------------------
Interface Rx Statistics:
   Unicast Uncontrolled Pkts: 8067779
   Multicast Uncontrolled Pkts: 14
   Broadcast Uncontrolled Pkts: 0
   Uncontrolled Pkts - Rx Drop: 0
   Uncontrolled Pkts - Rx Error: 0
   Unicast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Multicast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Broadcast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Controlled Pkts: 8056748
   Controlled Pkts - Rx Drop: N/A (N3K-C3636C-R not supported)
   Controlled Pkts - Rx Error: N/A (N3K-C3636C-R not supported)
   In-Octets Uncontrolled: 37641828280 bytes
   In-Octets Controlled: 37324295914 bytes
   Input rate for Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Input rate for Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Input rate for Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Input rate for Controlled Pkts: N/A (N3K-C3636C-R not supported)

Interface Tx Statistics:
   Unicast Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Multicast Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Broadcast Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Uncontrolled Pkts - Rx Drop: N/A (N3K-C3636C-R not supported)
   Uncontrolled Pkts - Rx Error: N/A (N3K-C3636C-R not supported)
   Unicast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Multicast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Broadcast Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Controlled Pkts: 8049279
   Controlled Pkts - Rx Drop: N/A (N3K-C3636C-R not supported)
   Controlled Pkts - Rx Error: N/A (N3K-C3636C-R not supported)
   Out-Octets Uncontrolled: N/A (N3K-C3636C-R not supported)
   Out-Octets Controlled: 37262189352 bytes
```

```
   Out-Octets Common: 37699748491 bytes
   Output rate for Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Output rate for Uncontrolled Pkts: N/A (N3K-C3636C-R not supported)
   Output rate for Controlled Pkts: N/A (N3K-C3636C-R not supported)
   Output rate for Controlled Pkts: N/A (N3K-C3636C-R not supported)

SECY Rx Statistics:
   Transform Error Pkts: N/A (N3K-C3636C-R not supported)
   Control Pkts: 0
   Untagged Pkts: N/A (N3K-C3636C-R not supported)
   No Tag Pkts: 0
   Bad Tag Pkts: 0
   No SCI Pkts: 0
   Unknown SCI Pkts: 0
   Tagged Control Pkts: N/A (N3K-C3636C-R not supported)

SECY Tx Statistics:
   Transform Error Pkts: N/A (N3K-C3636C-R not supported)
   Control Pkts: 0
   Untagged Pkts: N/A (N3K-C3636C-R not supported)

SAK Rx Statistics for AN [0]:
   Unchecked Pkts: 0
   Delayed Pkts: 0
   Late Pkts: 0
   OK Pkts: 8056748
   Invalid Pkts: 0
   Not Valid Pkts: 0
   Not-Using-SA Pkts: 0
   Unused-SA Pkts: 0
   Decrypted In-Octets: 36952542946 bytes
   Validated In-Octets: 0 bytes

SAK Tx Statistics for AN [0]:
   Encrypted Protected Pkts: 8049279
   Too Long Pkts: N/A (N3K-C3636C-R not supported)
   SA-not-in-use Pkts: N/A (N3K-C3636C-R not supported)
   Encrypted Protected Out-Octets: 36909704659 bytes

switch(config)#
```

Configuration Example for MACsec

The following example shows how to configure a user-defined MACsec policy and then apply the policy to interfaces:

```
switch(config)# macsec policy mpsr1
switch(config-macsec-policy)# cipher-suite GCM-AES-128
switch(config-macsec-policy)# key-server-priority 1
switch(config-macsec-policy)# window-size 1000
switch(config-macsec-policy)# conf-offset CONF-OFFSET-30
switch(config-macsec-policy)# security-policy must-secure
switch(config-macsec-policy)# sak-expiry-time 60
switch(config-macsec-policy)# include-icv-indicator

switch(config-macsec-policy)# interface e1/35-36
switch(config-if-range)# macsec keychain ksr policy mpsr1
switch(config-if-range)# show macsec mka session
Interface          Local-TxSCI                     # Peers            Status
Key-Server         Auth Mode
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/35       6c8b.d3db.e980/0001             1                  Secured
```

```
Yes                PRIMARY-PSK
Ethernet1/36       6c8b.d3db.e984/0001              1                  Secured
No                 PRIMARY-PSK
------------------ -------------------------------- ------------------ ------------------
------------------ ------------------

switch(config-if-range)# show macsec mka summary
Interface          Status   Cipher (Operational)   Key-Server   MACSEC-policy
       Keychain                        Fallback-keychain
------------------ -------- ---------------------- ------------
-------------------------------- --------------------------------
--------------------------------
Ethernet1/35       Secured  GCM-AES-128            Yes          mpsr1
       ksr                             no keychain
Ethernet1/36       Secured  GCM-AES-128            No           mpsr1
       ksr                             no keychain

switch(config-if-range)# show running-config macsec
!Command: show running-config macsec
!Running configuration last done at: Tue Dec 15 11:41:53 2020
!Time: Tue Dec 15 11:45:06 2020

version 10.1(1) Bios:version 01.14
feature macsec

macsec policy mpsr1
   cipher-suite GCM-AES-128
   key-server-priority 1
   window-size 1000
   conf-offset CONF-OFFSET-30
   sak-expiry-time 60
   include-icv-indicator

interface Ethernet1/35
   macsec keychain ksr policy mpsr1

interface Ethernet1/36
   macsec keychain ksr policy mpsr1
```

The following example shows how to configure a MACsec keychain and then add the system default MACsec policy to the interfaces:

```
switch(config)# key chain ksr macsec
switch(config-macseckeychain)# key 2006
switch(config-macseckeychain-macseckey)# key-octet-string
1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef cryptographic-algorithm
AES_256_CMAC
switch(config-macseckeychain-macseckey)# interface e1/35-36
switch(config-if-range)# macsec keychain ksr

switch(config-if-range)# show running-config macsec
!Command: show running-config macsec
!Running configuration last done at: Tue Dec 15 11:53:10 2020
!Time: Tue Dec 15 11:54:40 2020

version 10.1(1) Bios:version 01.14
feature macsec

interface Ethernet1/35
   macsec keychain ksr policy system-default-macsec-policy

interface Ethernet1/36
   macsec keychain ksr policy system-default-macsec-policy
```

```
switch(config-if-range)# show macsec mka summary
Interface          Status   Cipher (Operational)   Key-Server   MACSEC-policy
       Keychain                      Fallback-keychain
------------------ -------- ---------------------- ------------
--------------------------------- ---------------------------------
---------------------------------
Ethernet1/35       Secured  GCM-AES-XPN-256        Yes          system-default-macsec-policy
     ksr                           no keychain
Ethernet1/36       Secured  GCM-AES-XPN-256        No           system-default-macsec-policy
     ksr                           no keychain

switch(config-if-range)# show macsec mka session
Interface          Local-TxSCI                       # Peers            Status
Key-Server         Auth Mode
------------------ --------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/35       6c8b.d3db.e980/0001               1                  Secured
Yes                PRIMARY-PSK
Ethernet1/36       6c8b.d3db.e984/0001               1                  Secured
No                 PRIMARY-PSK
------------------ --------------------------------- ------------------ ------------------
------------------ ------------------

Total Number of Sessions : 2
   Secured Sessions : 2
   Pending Sessions : 0

switch(config-if-range)#
```

XML Examples

MACsec supports XML output for the following **show** commands for scripting purposes using | **xml**:

- **show key chain** *name* | **xml**
- **show macsec mka session** *interface interface slot/port* **details** |**xml**
- **show macsec mka statistics interface** *interface slot/port* |**xml**
- **show macsec mka summary** |**xml**
- **show macsec policy** *name* |**xml**
- **show macsec secy statistics interface** *interface slot/port* |**xml**
- **show running-config macsec** |**xml**

The following are example outputs for each of the preceding **show** commands:

Example 1: Displays the keychain configuration

```
switch(config)# show key chain "ksr" | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:rpm"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <key>
    <chain>
     <__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
      <keychain>ksr</keychain>
      <__XML__OPT_Cmd_rpm_show_keychain_cmd___readonly__>
```

```
        <__readonly__>
         <TABLE_keychain>
          <ROW_keychain>
           <chain_name>ksr</chain_name>
           <TABLE_key>
            <ROW_key>
             <key_id>2006</key_id>
<key_string>075e731f1a5c4f524f4cfb0d0629f212e62677147524054590f09595 1570a061e470c0c030604020c520c07055c5301155756085f5359761417591807141 60e0e</key_string>
             <crypto_algo>AES_256_CMAC</crypto_algo>
             <send_valid>true</send_valid>
            </ROW_key>
           </TABLE_key>
          </ROW_keychain>
         </TABLE_keychain>
        </__readonly__>
       </__XML__OPT_Cmd_rpm_show_keychain_cmd___readonly__>
      </__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
     </chain>
    </key>
   </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 2: Displays information about the MACsec MKA session for a specific interface

```
switch(config)# show macsec mka session interface e1/35 details | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:cts"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <macsec>
    <mka>
     <session>
      <__XML__OPT_Cmd_show_macsec_mka_session_interface>
       <interface>
        <__XML__INTF_ifname>
         <__XML__PARAM_value>
          <__XML__INTF_output>Ethernet1/35</__XML__INTF_output>
         </__XML__PARAM_value>
        </__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_show_macsec_mka_session_details>
        <details/>
        <__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
         <__readonly__>
          <TABLE_mka_session_details>
           <ROW_mka_session_details>
            <ifname>Ethernet1/35</ifname>
            <status>SECURED - Secured MKA Session with MACsec</status>
            <sci>6c8b.d3db.e980/0001</sci>
            <ssci>2</ssci>
            <port_id>2</port_id>
            <ckn>2006</ckn>
            <ca_auth_mode>PRIMARY-PSK</ca_auth_mode>
            <mi>5AABE0AB9CC867AB0FF40F7D</mi>
            <mn>3550</mn>
            <policy>system-default-macsec-policy</policy>
            <ks_prio>16</ks_prio>
            <keyserver>Yes</keyserver>
```

```
            <include_icv_indicator>No</include_icv_indicator>
            <cipher>GCM-AES-XPN-256</cipher>
            <cipher_operational>GCM-AES-XPN-256</cipher_operational>
            <window>148809600</window>
            <conf_offset>CONF-OFFSET-0</conf_offset>
            <conf_offset_operational>CONF-OFFSET-0</conf_offset_operational>
            <sak_status>Rx &amp; TX</sak_status>
            <sak_an>0</sak_an>
            <sak_ki>5AABE0AB9CC867AB0FF40F7D00000001</sak_ki>
            <sak_kn>1</sak_kn>
            <last_sak_rekey_time>11:53:25 pst Tue Dec 15 2020</last_sak_rekey_time>
            <peer_count>1</peer_count>
            <mac_addr>0180.c200.0003</mac_addr>
            <ether_type>0x888e</ether_type>
            <TABLE_mka_peer_status>
             <ROW_mka_peer_status>
              <peer_mi>27FC36C2BFAFBDBC65419A40</peer_mi>
              <rxsci>6c8b.d3db.e984/0001</rxsci>
              <icv_status>Match</icv_status>
              <last_rx_time>13:51:39 pst Tue Dec 15 2020</last_rx_time>
             </ROW_mka_peer_status>
            </TABLE_mka_peer_status>
           </ROW_mka_session_details>
          </TABLE_mka_session_details>
         </__readonly__>
        </__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
       </__XML__OPT_Cmd_show_macsec_mka_session_details>
      </__XML__OPT_Cmd_show_macsec_mka_session_interface>
     </session>
    </mka>
   </macsec>
  </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 3: Displays MACsec MKA statistics

```
switch(config)# show macsec mka statistics interface e1/29 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:cts"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <macsec>
    <mka>
     <statistics>
      <__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
       <interface>
        <__XML__INTF_ifname>
         <__XML__PARAM_value>
          <__XML__INTF_output>Ethernet1/29</__XML__INTF_output>
         </__XML__PARAM_value>
        </__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
        <__readonly__>
         <TABLE_mka_intf_stats>
          <ROW_mka_intf_stats>
           <ifname2>Ethernet1/29</ifname2>
           <TABLE_ca_stats>
            <ROW_ca_stats>
             <ca_stat_ckn>2002</ca_stat_ckn>
```

```
                    <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
                    <sa_stat_sak_generated>0</sa_stat_sak_generated>
                    <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
                    <sa_stat_sak_received>2</sa_stat_sak_received>
                    <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
                    <mkpdu_stat_mkpdu_tx>4335</mkpdu_stat_mkpdu_tx>
                    <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
                    <mkpdu_stat_mkpdu_rx>4335</mkpdu_stat_mkpdu_rx>
                    <mkpdu_stat_mkpdu_rx_distsak>2</mkpdu_stat_mkpdu_rx_distsak>
                   </ROW_ca_stats>
                  </TABLE_ca_stats>
                  <TABLE_idb_stats>
                   <ROW_idb_stats>
                    <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
                    <sa_stat_sak_generated>0</sa_stat_sak_generated>
                    <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
                    <sa_stat_sak_received>2</sa_stat_sak_received>
                    <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
                    <mkpdu_stat_mkpdu_tx>4335</mkpdu_stat_mkpdu_tx>
                    <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
                    <mkpdu_stat_mkpdu_rx>4335</mkpdu_stat_mkpdu_rx>
                    <mkpdu_stat_mkpdu_rx_distsak>2</mkpdu_stat_mkpdu_rx_distsak>
                    <idb_stat_mkpdu_tx_success>8666</idb_stat_mkpdu_tx_success>
                    <idb_stat_mkpdu_tx_fail>0</idb_stat_mkpdu_tx_fail>
                    <idb_stat_mkpdu_tx_pkt_build_fail>0</idb_stat_mkpdu_tx_pkt_build_fail>
                    <idb_stat_mkpdu_no_tx_on_intf_down>0</idb_stat_mkpdu_no_tx_on_intf_down>
                    <idb_stat_mkpdu_no_rx_on_intf_down>0</idb_stat_mkpdu_no_rx_on_intf_down>
                    <idb_stat_mkpdu_rx_ca_notfound>0</idb_stat_mkpdu_rx_ca_notfound>
                    <idb_stat_mkpdu_rx_error>0</idb_stat_mkpdu_rx_error>
                    <idb_stat_mkpdu_rx_success>8666</idb_stat_mkpdu_rx_success>
<idb_stat_mkpdu_failure_rx_integrity_check_error>0</idb_stat_mkpdu_failure_rx_integrity_check_error>
<idb_stat_mkpdu_failure_invalid_peer_mn_error>0</idb_stat_mkpdu_failure_invalid_peer_mn_error>
<idb_stat_mkpdu_failure_nonrecent_peerlist_mn_error>0</idb_stat_mkpdu_failure_nonrecent_peerlist_mn_error>
<idb_stat_mkpdu_failure_sakuse_kn_mismatch_error>0</idb_stat_mkpdu_failure_sakuse_kn_mismatch_error>
<idb_stat_mkpdu_failure_sakuse_rx_not_set_error>0</idb_stat_mkpdu_failure_sakuse_rx_not_set_error>
<idb_stat_mkpdu_failure_sakuse_key_mi_mismatch_error>0</idb_stat_mkpdu_failure_sakuse_key_mi_mismatch_error>
<idb_stat_mkpdu_failure_sakuse_an_not_in_use_error>0</idb_stat_mkpdu_failure_sakuse_an_not_in_use_error>
<idb_stat_mkpdu_failure_sakuse_ks_rx_tx_not_set_error>0</idb_stat_mkpdu_failure_sakuse_ks_rx_tx_not_set_error>
<idb_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>0</idb_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>
<idb_stat_mkpdu_failure_sakuse_eapol_destmac_mismatch_error>0</idb_stat_mkpdu_failure_sakuse_eapol_destmac_mismatch_error>
<idb_stat_sak_failure_sak_generate_error>0</idb_stat_sak_failure_sak_generate_error>
<idb_stat_sak_failure_hash_generate_error>0</idb_stat_sak_failure_hash_generate_error>
```

```
<idb_stat_sak_failure_sak_encryption_error>0</idb_stat_sak_failure_sak_encryption_error>

<idb_stat_sak_failure_sak_decryption_error>0</idb_stat_sak_failure_sak_decryption_error>

<idb_stat_sak_failure_ick_derivation_error>0</idb_stat_sak_failure_ick_derivation_error>

<idb_stat_sak_failure_kek_derivation_error>0</idb_stat_sak_failure_kek_derivation_error>

<idb_stat_sak_failure_invalid_macsec_capability_error>0</idb_stat_sak_failure_invalid_macsec_capability_error>


<idb_stat_macsec_failure_rx_sa_create_error>0</idb_stat_macsec_failure_rx_sa_create_error>

<idb_stat_macsec_failure_tx_sa_create_error>0</idb_stat_macsec_failure_tx_sa_create_error>
            </ROW_idb_stats>
           </TABLE_idb_stats>
          </ROW_mka_intf_stats>
         </TABLE_mka_intf_stats>
        </__readonly__>
       </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
      </__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
     </statistics>
    </mka>
   </macsec>
  </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 4: Displays the MACsec MKA configuration

```
switch(config)# show macsec mka summary | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:cts"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <macsec>
    <mka>
     <__XML__OPT_Cmd_some_macsec_summary>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
        <TABLE_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet1/29</ifname>
          <status>Secured</status>
          <cipher>GCM-AES-128</cipher>
          <keyserver>No</keyserver>
          <policy>mpd1</policy>
          <keychain>kd</keychain>
          <fallback_keychain>fbkd</fallback_keychain>
         </ROW_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet1/30</ifname>
          <status>Secured</status>
          <cipher>GCM-AES-128</cipher>
          <keyserver>No</keyserver>
          <policy>mpd2</policy>
          <keychain>kd</keychain>
          <fallback_keychain>fbkd</fallback_keychain>
         </ROW_mka_summary>
         <ROW_mka_summary>
```

```
            <ifname>Ethernet1/31</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-128</cipher>
            <keyserver>Yes</keyserver>
            <policy>mps1</policy>
            <keychain>ks</keychain>
            <fallback_keychain>fbks</fallback_keychain>
           </ROW_mka_summary>
           <ROW_mka_summary>
            <ifname>Ethernet1/32</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-128</cipher>
            <keyserver>Yes</keyserver>
            <policy>mps2</policy>
            <keychain>ks</keychain>
            <fallback_keychain>fbks</fallback_keychain>
           </ROW_mka_summary>
           <ROW_mka_summary>
            <ifname>Ethernet1/33</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-128</cipher>
            <keyserver>Yes</keyserver>
            <policy>mpsr1</policy>
            <keychain>ksr</keychain>
            <fallback_keychain>fbksr</fallback_keychain>
           </ROW_mka_summary>
           <ROW_mka_summary>
            <ifname>Ethernet1/34</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-128</cipher>
            <keyserver>Yes</keyserver>
            <policy>mpsr2</policy>
            <keychain>ksr</keychain>
            <fallback_keychain>fbksr</fallback_keychain>
           </ROW_mka_summary>
           <ROW_mka_summary>
            <ifname>Ethernet1/35</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-XPN-256</cipher>
            <keyserver>Yes</keyserver>
            <policy>system-default-macsec-policy</policy>
            <keychain>ksr</keychain>
            <fallback_keychain>no keychain</fallback_keychain>
           </ROW_mka_summary>
           <ROW_mka_summary>
            <ifname>Ethernet1/36</ifname>
            <status>Secured</status>
            <cipher>GCM-AES-XPN-256</cipher>
            <keyserver>No</keyserver>
            <policy>system-default-macsec-policy</policy>
            <keychain>ksr</keychain>
            <fallback_keychain>no keychain</fallback_keychain>
           </ROW_mka_summary>
          </TABLE_mka_summary>
         </__readonly__>
        </__XML__OPT_Cmd_some_macsec___readonly__>
       </__XML__OPT_Cmd_some_macsec_summary>
      </mka>
     </macsec>
    </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 5: Displays the configuration for a specific MACsec policy

```
switch(config)# show macsec policy mpsr1 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:cts"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <macsec>
    <policy>
     <__XML__OPT_Cmd_show_macsec_policy_policy_name>
      <policy_name>mpsr1</policy_name>
      <__XML__OPT_Cmd_show_macsec_policy___readonly__>
       <__readonly__>
        <TABLE_macsec_policy>
         <ROW_macsec_policy>
          <name>mpsr1</name>
          <cipher_suite>GCM-AES-128</cipher_suite>
          <keyserver_priority>1</keyserver_priority>
          <window_size>1000</window_size>
          <conf_offset>30</conf_offset>
          <security_policy>should-secure</security_policy>
          <sak-expiry-time>60</sak-expiry-time>
          <include_icv_indicator>TRUE</include_icv_indicator>
         </ROW_macsec_policy>
        </TABLE_macsec_policy>
       </__readonly__>
      </__XML__OPT_Cmd_show_macsec_policy___readonly__>
     </__XML__OPT_Cmd_show_macsec_policy_policy_name>
    </policy>
   </macsec>
  </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 6: Displays MACsec Security statistics

```
switch(config)# show macsec secy statistics interface e1/29 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns="http://www.cisco.com/nxos:1.0:cts"
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0">
<nf:data>
  <show>
   <macsec>
    <secy>
     <statistics>
      <__XML__OPT_Cmd_some_macsec_secy_statistics_interface>
       <interface>
        <__XML__INTF_ifname>
         <__XML__PARAM_value>
          <__XML__INTF_output>Ethernet1/29</__XML__INTF_output>
         </__XML__PARAM_value>
        </__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
        <__readonly__>
         <TABLE_statistics>
          <ROW_statistics>
           <ifname2>Ethernet1/29</ifname2>
           <in_pkts_unicast_uncontrolled>6536205587</in_pkts_unicast_uncontrolled>
           <in_pkts_multicast_uncontrolled>10775</in_pkts_multicast_uncontrolled>
           <in_pkts_broadcast_uncontrolled>0</in_pkts_broadcast_uncontrolled>
```

```
            <in_rx_drop_pkts_uncontrolled>0</in_rx_drop_pkts_uncontrolled>
            <in_rx_err_pkts_uncontrolled>0</in_rx_err_pkts_uncontrolled>
            <in_pkts_unicast_controlled>N/A (N3K-C3636C-R not
supported)</in_pkts_unicast_controlled>
            <in_pkts_multicast_controlled>N/A (N3K-C3636C-R not
supported)</in_pkts_multicast_controlled>
            <in_pkts_broadcast_controlled>N/A (N3K-C3636C-R not
supported)</in_pkts_broadcast_controlled>
            <in_pkts_controlled>5173107800</in_pkts_controlled>
            <in_rx_drop_pkts_controlled>N/A (N3K-C3636C-R not
supported)</in_rx_drop_pkts_controlled>
            <in_rx_err_pkts_controlled>N/A (N3K-C3636C-R not
supported)</in_rx_err_pkts_controlled>
            <in_octets_uncontrolled>30491280431357</in_octets_uncontrolled>
            <in_octets_controlled>23935220809548</in_octets_controlled>
            <input_rate_uncontrolled_pps>N/A (N3K-C3636C-R not
supported)</input_rate_uncontrolled_pps>
            <input_rate_uncontrolled_bps>N/A (N3K-C3636C-R not
supported)</input_rate_uncontrolled_bps>
            <input_rate_controlled_pps>N/A (N3K-C3636C-R not
supported)</input_rate_controlled_pps>
            <input_rate_controlled_bps>N/A (N3K-C3636C-R not
supported)</input_rate_controlled_bps>
            <out_pkts_unicast_uncontrolled>N/A (N3K-C3636C-R not
supported)</out_pkts_unicast_uncontrolled>
            <out_pkts_multicast_uncontrolled>N/A (N3K-C3636C-R not
supported)</out_pkts_multicast_uncontrolled>
            <out_pkts_broadcast_uncontrolled>N/A (N3K-C3636C-R not
supported)</out_pkts_broadcast_uncontrolled>
            <out_rx_drop_pkts_uncontrolled>N/A (N3K-C3636C-R not
supported)</out_rx_drop_pkts_uncontrolled>
            <out_rx_err_pkts_uncontrolled>N/A (N3K-C3636C-R not
supported)</out_rx_err_pkts_uncontrolled>
            <out_pkts_unicast_controlled>N/A (N3K-C3636C-R not
supported)</out_pkts_unicast_controlled>
            <out_pkts_multicast_controlled>N/A (N3K-C3636C-R not
supported)</out_pkts_multicast_controlled>
            <out_pkts_broadcast_controlled>N/A (N3K-C3636C-R not
supported)</out_pkts_broadcast_controlled>
            <out_pkts_controlled>5173113173</out_pkts_controlled>
            <out_rx_drop_pkts_controlled>N/A (N3K-C3636C-R not
supported)</out_rx_drop_pkts_controlled>
            <out_rx_err_pkts_controlled>N/A (N3K-C3636C-R not
supported)</out_rx_err_pkts_controlled>
          <out_octets_uncontrolled>N/A (N3K-C3636C-R not supported)</out_octets_uncontrolled>

            <out_octets_controlled>23946219872208</out_octets_controlled>
            <out_octets_common>30664229104600</out_octets_common>
            <output_rate_uncontrolled_pps>N/A (N3K-C3636C-R not
supported)</output_rate_uncontrolled_pps>
            <output_rate_uncontrolled_bps>N/A (N3K-C3636C-R not
supported)</output_rate_uncontrolled_bps>
            <output_rate_controlled_pps>N/A (N3K-C3636C-R not
supported)</output_rate_controlled_pps>
            <output_rate_controlled_bps>N/A (N3K-C3636C-R not
supported)</output_rate_controlled_bps>
          <in_pkts_transform_error>N/A (N3K-C3636C-R not supported)</in_pkts_transform_error>

            <in_pkts_control>0</in_pkts_control>
            <in_pkts_untagged>N/A (N3K-C3636C-R not supported)</in_pkts_untagged>
            <in_pkts_no_tag>0</in_pkts_no_tag>
            <in_pkts_badtag>0</in_pkts_badtag>
            <in_pkts_no_sci>0</in_pkts_no_sci>
            <in_pkts_unknown_sci>0</in_pkts_unknown_sci>
```

```
           <in_pkts_tagged_ctrl>N/A (N3K-C3636C-R not supported)</in_pkts_tagged_ctrl>
           <out_pkts_transform_error>N/A (N3K-C3636C-R not
supported)</out_pkts_transform_error>
           <out_pkts_control>0</out_pkts_control>
           <out_pkts_untagged>N/A (N3K-C3636C-R not supported)</out_pkts_untagged>
           <TABLE_rx_sa_an>
            <ROW_rx_sa_an>
             <rx_sa_an>2</rx_sa_an>
             <in_pkts_unchecked>0</in_pkts_unchecked>
             <in_pkts_delayed>0</in_pkts_delayed>
             <in_pkts_late>0</in_pkts_late>
             <in_pkts_ok>1951781408</in_pkts_ok>
             <in_pkts_invalid>0</in_pkts_invalid>
             <in_pkts_not_valid>0</in_pkts_not_valid>
             <in_pkts_not_using_sa>0</in_pkts_not_using_sa>
             <in_pkts_unused_sa>0</in_pkts_unused_sa>
             <in_octets_decrypted>8952613134278</in_octets_decrypted>
             <in_octets_validated>0</in_octets_validated>
            </ROW_rx_sa_an>
           </TABLE_rx_sa_an>
           <TABLE_tx_sa_an>
            <ROW_tx_sa_an>
             <tx_sa_an>2</tx_sa_an>
             <out_pkts_encrypted_protected>1951773387</out_pkts_encrypted_protected>
             <out_pkts_too_long>N/A (N3K-C3636C-R not supported)</out_pkts_too_long>
           <out_pkts_sa_not_inuse>N/A (N3K-C3636C-R not supported)</out_pkts_sa_not_inuse>

            <out_octets_encrypted_protected>8952606203313</out_octets_encrypted_protected>

            </ROW_tx_sa_an>
           </TABLE_tx_sa_an>
          </ROW_statistics>
         </TABLE_statistics>
        </__readonly__>
       </__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
      </__XML__OPT_Cmd_some_macsec_secy_statistics_interface>
     </statistics>
    </secy>
   </macsec>
  </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

Example 7: Displays the running configuration information for MACsec

```
switch(config)# show running-config macsec | xml

!Command: show running-config macsec
!Running configuration last done at: Tue Dec 15 11:53:10 2020
!Time: Tue Dec 15 13:58:58 2020

version 10.1(1) Bios:version 01.14
****************************************
This may take time. Please be patient.
****************************************
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://www.cisco.com/nxos:10.1.1.:configure_"
xmlns:m="http://www.cisco.com/nxos:10.1.1.:_exec"
xmlns:m1="http://www.cisco.com/nxos:10.1.1.:configure__macsec-policy"
xmlns:m2="http://www.cisco.com/nxos:10.1.1.:configure__if-ethernet-all" message-id="1">
  <nf:get-config>
```

```
<nf:source>
  <nf:running/>
</nf:source>
<nf:filter>
  <m:configure>
    <m:terminal>
      <feature>
        <macsec/>
      </feature>
      <macsec>
        <policy>
          <__XML__PARAM__policy_name>
            <__XML__value>mpd1</__XML__value>
            <m1:cipher-suite>
              <m1:__XML__PARAM__suite>
                <m1:__XML__value>GCM-AES-128</m1:__XML__value>
              </m1:__XML__PARAM__suite>
            </m1:cipher-suite>
            <m1:conf-offset>
              <m1:__XML__PARAM__offset>
                <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
              </m1:__XML__PARAM__offset>
            </m1:conf-offset>
          </__XML__PARAM__policy_name>
        </policy>
      </macsec>
      <macsec>
        <policy>
          <__XML__PARAM__policy_name>
            <__XML__value>mpd2</__XML__value>
            <m1:cipher-suite>
              <m1:__XML__PARAM__suite>
                <m1:__XML__value>GCM-AES-128</m1:__XML__value>
              </m1:__XML__PARAM__suite>
            </m1:cipher-suite>
            <m1:conf-offset>
              <m1:__XML__PARAM__offset>
                <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
              </m1:__XML__PARAM__offset>
            </m1:conf-offset>
            <m1:security-policy>
              <m1:__XML__PARAM__policy>
                <m1:__XML__value>must-secure</m1:__XML__value>
              </m1:__XML__PARAM__policy>
            </m1:security-policy>
          </__XML__PARAM__policy_name>
        </policy>
      </macsec>
      <macsec>
        <policy>
          <__XML__PARAM__policy_name>
            <__XML__value>mps1</__XML__value>
            <m1:cipher-suite>
              <m1:__XML__PARAM__suite>
                <m1:__XML__value>GCM-AES-128</m1:__XML__value>
             </m1:__XML__PARAM__suite>
            </m1:cipher-suite>
            <m1:key-server-priority>
              <m1:__XML__PARAM__pri>
                <m1:__XML__value>1</m1:__XML__value>
              </m1:__XML__PARAM__pri>
            </m1:key-server-priority>
            <m1:conf-offset>
              <m1:__XML__PARAM__offset>
```

```
            <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
          </m1:__XML__PARAM__offset>
        </m1:conf-offset>
        <m1:sak-expiry-time>
          <m1:__XML__PARAM__ts>
            <m1:__XML__value>60</m1:__XML__value>
          </m1:__XML__PARAM__ts>
        </m1:sak-expiry-time>
        <m1:include-icv-indicator/>
      </__XML__PARAM__policy_name>
    </policy>
  </macsec>
  <macsec>
    <policy>
      <__XML__PARAM__policy_name>
        <__XML__value>mps2</__XML__value>
        <m1:cipher-suite>
          <m1:__XML__PARAM__suite>
            <m1:__XML__value>GCM-AES-128</m1:__XML__value>
          </m1:__XML__PARAM__suite>
        </m1:cipher-suite>
        <m1:key-server-priority>
          <m1:__XML__PARAM__pri>
            <m1:__XML__value>1</m1:__XML__value>
          </m1:__XML__PARAM__pri>
        </m1:key-server-priority>
        <m1:window-size>
          <m1:__XML__PARAM__size>
            <m1:__XML__value>1000</m1:__XML__value>
          </m1:__XML__PARAM__size>
        </m1:window-size>
        <m1:conf-offset>
          <m1:__XML__PARAM__offset>
            <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
          </m1:__XML__PARAM__offset>
        </m1:conf-offset>
        <m1:security-policy>
          <m1:__XML__PARAM__policy>
            <m1:__XML__value>must-secure</m1:__XML__value>
          </m1:__XML__PARAM__policy>
        </m1:security-policy>
        <m1:sak-expiry-time>
          <m1:__XML__PARAM__ts>
            <m1:__XML__value>60</m1:__XML__value>
          </m1:__XML__PARAM__ts>
        </m1:sak-expiry-time>
        <m1:include-icv-indicator/>
      </__XML__PARAM__policy_name>
    </policy>
  </macsec>
  <macsec>
    <policy>
      <__XML__PARAM__policy_name>
        <__XML__value>mpsr1</__XML__value>
        <m1:cipher-suite>
          <m1:__XML__PARAM__suite>
            <m1:__XML__value>GCM-AES-128</m1:__XML__value>
          </m1:__XML__PARAM__suite>
        </m1:cipher-suite>
        <m1:key-server-priority>
          <m1:__XML__PARAM__pri>
            <m1:__XML__value>1</m1:__XML__value>
          </m1:__XML__PARAM__pri>
        </m1:key-server-priority>
```

```
      <m1:window-size>
        <m1:__XML__PARAM__size>
          <m1:__XML__value>1000</m1:__XML__value>
        </m1:__XML__PARAM__size>
      </m1:window-size>
      <m1:conf-offset>
        <m1:__XML__PARAM__offset>
          <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
        </m1:__XML__PARAM__offset>
      </m1:conf-offset>
      <m1:sak-expiry-time>
        <m1:__XML__PARAM__ts>
          <m1:__XML__value>60</m1:__XML__value>
        </m1:__XML__PARAM__ts>
      </m1:sak-expiry-time>
      <m1:include-icv-indicator/>
    </__XML__PARAM__policy_name>
  </policy>
</macsec>
<macsec>
  <policy>
    <__XML__PARAM__policy_name>
      <__XML__value>mpsr2</__XML__value>
      <m1:cipher-suite>
        <m1:__XML__PARAM__suite>
          <m1:__XML__value>GCM-AES-128</m1:__XML__value>
        </m1:__XML__PARAM__suite>
      </m1:cipher-suite>
      <m1:key-server-priority>
        <m1:__XML__PARAM__pri>
          <m1:__XML__value>1</m1:__XML__value>
        </m1:__XML__PARAM__pri>
      </m1:key-server-priority>
      <m1:window-size>
        <m1:__XML__PARAM__size>
          <m1:__XML__value>1000</m1:__XML__value>
        </m1:__XML__PARAM__size>
      </m1:window-size>
      <m1:conf-offset>
        <m1:__XML__PARAM__offset>
          <m1:__XML__value>CONF-OFFSET-30</m1:__XML__value>
        </m1:__XML__PARAM__offset>
      </m1:conf-offset>
      <m1:security-policy>
        <m1:__XML__PARAM__policy>
          <m1:__XML__value>must-secure</m1:__XML__value>
        </m1:__XML__PARAM__policy>
      </m1:security-policy>
      <m1:sak-expiry-time>
        <m1:__XML__PARAM__ts>
          <m1:__XML__value>60</m1:__XML__value>
        </m1:__XML__PARAM__ts>
      </m1:sak-expiry-time>
      <m1:include-icv-indicator/>
    </__XML__PARAM__policy_name>
  </policy>
</macsec>
<interface>
  <__XML__PARAM__interface>
    <__XML__value>Ethernet1/29</__XML__value>
    <m2:macsec>
      <m2:keychain>
        <m2:__XML__PARAM__keychain_name>
          <m2:__XML__value>kd</m2:__XML__value>
```

```
            <m2:policy>
              <m2:__XML__PARAM__policy_name>
                <m2:__XML__value>mpd1</m2:__XML__value>
                <m2:fallback-keychain>
                  <m2:__XML__PARAM__fallback_kc_name>
                    <m2:__XML__value>fbkd</m2:__XML__value>
                  </m2:__XML__PARAM__fallback_kc_name>
                </m2:fallback-keychain>
              </m2:__XML__PARAM__policy_name>
            </m2:policy>
          </m2:__XML__PARAM__keychain_name>
        </m2:keychain>
      </m2:macsec>
    </__XML__PARAM__interface>
  </interface>
  <interface>
    <__XML__PARAM__interface>
      <__XML__value>Ethernet1/30</__XML__value>
      <m2:macsec>
        <m2:keychain>
          <m2:__XML__PARAM__keychain_name>
            <m2:__XML__value>kd</m2:__XML__value>
            <m2:policy>
              <m2:__XML__PARAM__policy_name>
                <m2:__XML__value>mpd2</m2:__XML__value>
                <m2:fallback-keychain>
                  <m2:__XML__PARAM__fallback_kc_name>
                    <m2:__XML__value>fbkd</m2:__XML__value>
                  </m2:__XML__PARAM__fallback_kc_name>
                </m2:fallback-keychain>
              </m2:__XML__PARAM__policy_name>
            </m2:policy>
          </m2:__XML__PARAM__keychain_name>
        </m2:keychain>
      </m2:macsec>
    </__XML__PARAM__interface>
  </interface>
  <interface>
    <__XML__PARAM__interface>
      <__XML__value>Ethernet1/31</__XML__value>
      <m2:macsec>
        <m2:keychain>
          <m2:__XML__PARAM__keychain_name>
            <m2:__XML__value>ks</m2:__XML__value>
            <m2:policy>
              <m2:__XML__PARAM__policy_name>
                <m2:__XML__value>mps1</m2:__XML__value>
                <m2:fallback-keychain>
                  <m2:__XML__PARAM__fallback_kc_name>
                    <m2:__XML__value>fbks</m2:__XML__value>
                  </m2:__XML__PARAM__fallback_kc_name>
                </m2:fallback-keychain>
              </m2:__XML__PARAM__policy_name>
            </m2:policy>
          </m2:__XML__PARAM__keychain_name>
        </m2:keychain>
      </m2:macsec>
    </__XML__PARAM__interface>
  </interface>
  <interface>
    <__XML__PARAM__interface>
      <__XML__value>Ethernet1/32</__XML__value>
      <m2:macsec>
        <m2:keychain>
```

```
        <m2:__XML__PARAM__keychain_name>
          <m2:__XML__value>ks</m2:__XML__value>
          <m2:policy>
            <m2:__XML__PARAM__policy_name>
              <m2:__XML__value>mps2</m2:__XML__value>
              <m2:fallback-keychain>
                <m2:__XML__PARAM__fallback_kc_name>
                  <m2:__XML__value>fbks</m2:__XML__value>
                </m2:__XML__PARAM__fallback_kc_name>
              </m2:fallback-keychain>
            </m2:__XML__PARAM__policy_name>
          </m2:policy>
        </m2:__XML__PARAM__keychain_name>
      </m2:keychain>
    </m2:macsec>
  </__XML__PARAM__interface>
</interface>
<interface>
  <__XML__PARAM__interface>
    <__XML__value>Ethernet1/33</__XML__value>
    <m2:macsec>
      <m2:keychain>
        <m2:__XML__PARAM__keychain_name>
          <m2:__XML__value>ksr</m2:__XML__value>
          <m2:policy>
            <m2:__XML__PARAM__policy_name>
              <m2:__XML__value>mpsr1</m2:__XML__value>
              <m2:fallback-keychain>
                <m2:__XML__PARAM__fallback_kc_name>
                  <m2:__XML__value>fbksr</m2:__XML__value>
                </m2:__XML__PARAM__fallback_kc_name>
              </m2:fallback-keychain>
            </m2:__XML__PARAM__policy_name>
          </m2:policy>
        </m2:__XML__PARAM__keychain_name>
      </m2:keychain>
    </m2:macsec>
  </__XML__PARAM__interface>
</interface>
<interface>
  <__XML__PARAM__interface>
    <__XML__value>Ethernet1/34</__XML__value>
    <m2:macsec>
      <m2:keychain>
        <m2:__XML__PARAM__keychain_name>
          <m2:__XML__value>ksr</m2:__XML__value>
          <m2:policy>
            <m2:__XML__PARAM__policy_name>
              <m2:__XML__value>mpsr2</m2:__XML__value>
              <m2:fallback-keychain>
                <m2:__XML__PARAM__fallback_kc_name>
                  <m2:__XML__value>fbksr</m2:__XML__value>
                </m2:__XML__PARAM__fallback_kc_name>
              </m2:fallback-keychain>
            </m2:__XML__PARAM__policy_name>
          </m2:policy>
        </m2:__XML__PARAM__keychain_name>
      </m2:keychain>
    </m2:macsec>
  </__XML__PARAM__interface>
</interface>
<interface>
  <__XML__PARAM__interface>
    <__XML__value>Ethernet1/35</__XML__value>
```

```
                <m2:macsec>
                  <m2:keychain>
                    <m2:__XML__PARAM__keychain_name>
                      <m2:__XML__value>ksr</m2:__XML__value>
                      <m2:policy>
                        <m2:__XML__PARAM__policy_name>
                          <m2:__XML__value>system-default-macsec-policy</m2:__XML__value>
                        </m2:__XML__PARAM__policy_name>
                      </m2:policy>
                    </m2:__XML__PARAM__keychain_name>
                  </m2:keychain>
                </m2:macsec>
              </__XML__PARAM__interface>
            </interface>
            <interface>
              <__XML__PARAM__interface>
                <__XML__value>Ethernet1/36</__XML__value>
                <m2:macsec>
                  <m2:keychain>
                    <m2:__XML__PARAM__keychain_name>
                      <m2:__XML__value>ksr</m2:__XML__value>
                      <m2:policy>
                        <m2:__XML__PARAM__policy_name>
                          <m2:__XML__value>system-default-macsec-policy</m2:__XML__value>
                        </m2:__XML__PARAM__policy_name>
                      </m2:policy>
                    </m2:__XML__PARAM__keychain_name>
                  </m2:keychain>
                </m2:macsec>
              </__XML__PARAM__interface>
            </interface>
          </m:terminal>
        </m:configure>
      </nf:filter>
    </nf:get-config>
</nf:rpc>
]]>]]>

switch(config)#
```

MIBs

MACsec supports the following MIBs:

- IEEE8021-SECY-MIB
- CISCO-SECY-EXT-MIB

Related Documentation

Related Topic	Document Title
Keychain management	Cisco Nexus 3600 Series NX-OS Security Configuration Guide
System messages	Cisco Nexus 3600 Series NX-OS System Messages References

INDEX

E

G

H

I

L

M

P

R

S

T

U

V