Cisco Nexus 5500 Series NX-OS Security Configuration Guide, Release 7.x

First Published: 2014-01-29

Last Modified: 2020-05-07

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

Text Part Number: OL-30897-01

CONTENTS

Preface

The preface contains the following sections:

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions

Note As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

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.

Convention	Description
[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.
`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.

This document uses the following conventions:

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

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

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: .

We appreciate your feedback.

Communications, Services, and Additional Information

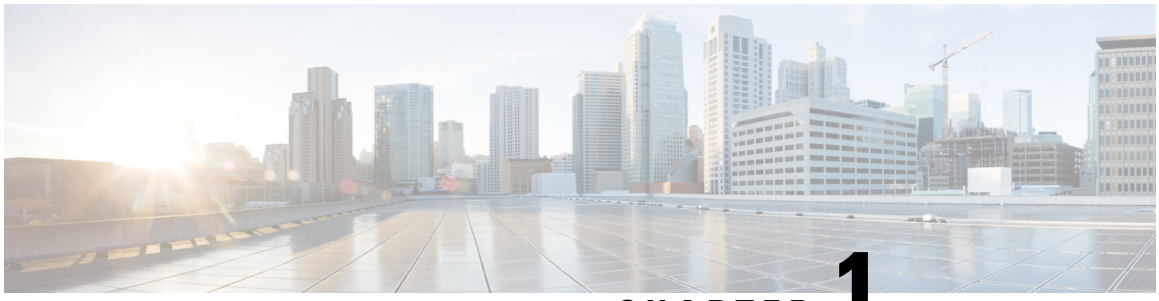
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.

- 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



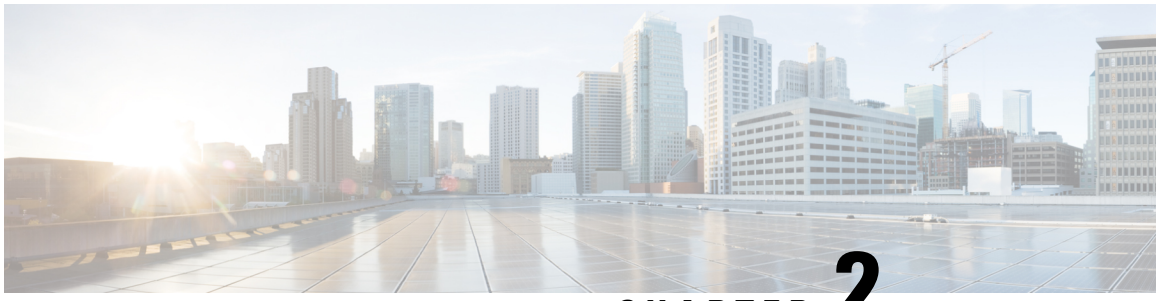
CHAPTER 1

New and Changed Information

The following table provides an overview of the significant changes made to this configuration guide. The table does not provide an exhaustive list of all changes made to this guide or all new features in a particular release.

Table 1: New and Changed Information

Feature	Description	Release	Where Documented
TCAM Carving	Enhancements to the TCAM carving feature to support reload of a switch after committing a template.	7.1(4)N1(1)	Configuring TCAM Carving
Port security	Minor enhancements to the port security feature.	7.1(4)N1(1)	Configuring Port Security
Object Group ACLs	Added the support for the object group ACLs.	7.3(0)N1(1)	Configuring Access Control Lists
Login Block Per User	Added support for login block per user.	7.3(0)N1(1)	Configuring Authentication, Authorization, and Accounting
Login Block Per User	Added support for login block per user.	7.3(0)N1(1)	Configuring Authentication, Authorization, and Accounting
Ternary Content-Addressable Memory (TCAM) carving feature.	The Ternary Content-Addressable Memory (TCAM) carving feature uses a template-based approach that enables you to modify the default region sizes of the TCAM.	7.0(0)N1(1)	Configuring TCAM Carving



CHAPTER 2

Overview

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

Authentication, Authorization, and Accounting

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

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

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

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

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

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

Related Topics

Configuring AAA

RADIUS and TACACS+ Security Protocols

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

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

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

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

Related Topics

Configuring RADIUS
Configuring TACACS+, on page 51

SSH and Telnet

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

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

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

Related Topics

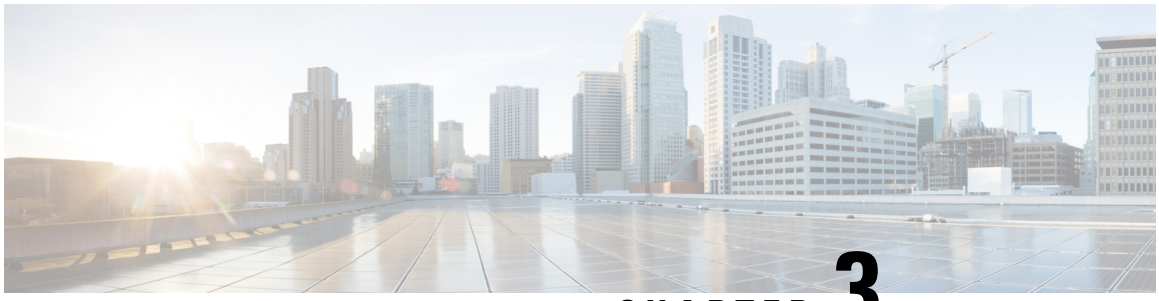
Configuring SSH and Telnet, on page 67

IP ACLs

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

Related Topics

Configuring IP ACLs



CHAPTER 3

Configuring FIPS

The Federal Information Processing Standards (FIPS) Publication 140-2, Security Requirements for Cryptographic Modules, details the U.S. government requirements for cryptographic modules. FIPS 140-2 specifies that a cryptographic module should be a set of hardware, software, firmware, or some combination that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary.

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

This chapter includes the following sections:

Configuration Guidelines

Follow these guidelines before enabling FIPS mode:

- Make your passwords a minimum of eight characters in length.
- Disable Telnet. Users should log in using SSH only.
- Disable remote authentication through RADIUS/TACACS+. Only users local to the switch can be authenticated.
- Disable SNMP v1 and v2. Any existing user accounts on the switch that have been configured for SNMPv3 should be configured only with SHA for authentication and AES/3DES for privacy.
- Disable VRRP.
- Do not configure FIPS and IPsec together on a switch. With FIPS enabled, if you configure IKE, then FCIP links will not come up.
- Delete all SSH Server RSA1 keypairs.
- Do not configure FIPS and RADIUS together on a switch.
- FIPS cannot function when RADIUS (MD5) is enabled. Hence you need to note the following:

Before you enable FIPS you need to disable RADIUS or select other authentication protocol other than MD5.

Before you enable RADIUS you need to disable FIPS if you need to use the RADIUS (MD5) authentication protocol.

Enabling FIPS Mode

To enable FIPS mode, follow these steps:

Procedure

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

Displaying FIPS Status

To view FIPS status, enter the **show fips status** command.

FIPS Self Tests

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

Note FIPS power-up self-tests automatically run when FIPS mode is enabled by entering the fips mode enable command. A switch is in FIPS mode only after all self-tests are successfully completed. If any of the self-tests fail, then the switch is rebooted.

Power-up self-tests run immediately after FIPS mode is enabled. A cryptographic algorithm test using a known answer must be run for all cryptographic functions for each FIPS 140-2-approved cryptographic algorithm implemented on the Cisco Nexus 5500 and 5600 Family.

Using a known-answer test (KAT), a cryptographic algorithm is run on data for which the correct output is already known, and then the calculated output is compared to the previously generated output. If the calculated output does not equal the known answer, the known-answer test fails.

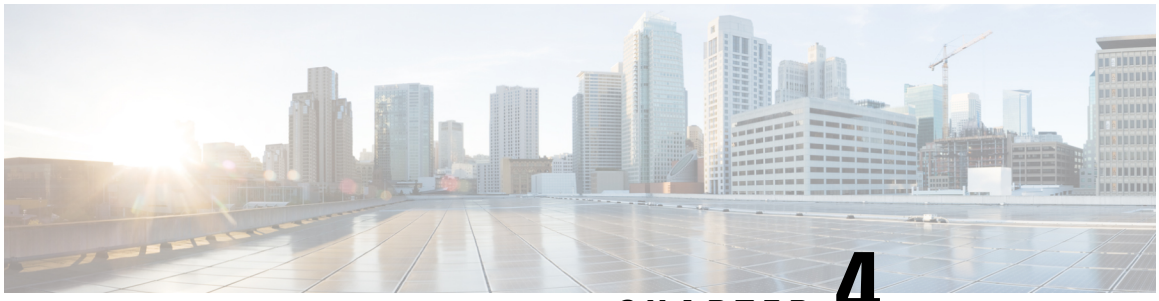
Conditional self-tests must be run when an applicable security function or operation is invoked. Unlike the power-up self-tests, conditional self-tests are executed each time their associated function is accessed.

Conditional self-tests include the following:

- Pair-wise consistency test—This test is run when a public-private keypair is generated

  .
- Continuous random number generator test—This test is run when a random number is generated.

Both of these tests automatically run when a switch is in FIPS mode.



CHAPTER 4

Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

Information About AAA

AAA Security Services

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

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

AAA security provides the following services:

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

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

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

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

Benefits of Using AAA

AAA provides the following benefits:

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

Remote AAA Services

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

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

AAA Server Groups

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

AAA Service Configuration Options

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

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

- User management session accounting

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

Table 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.

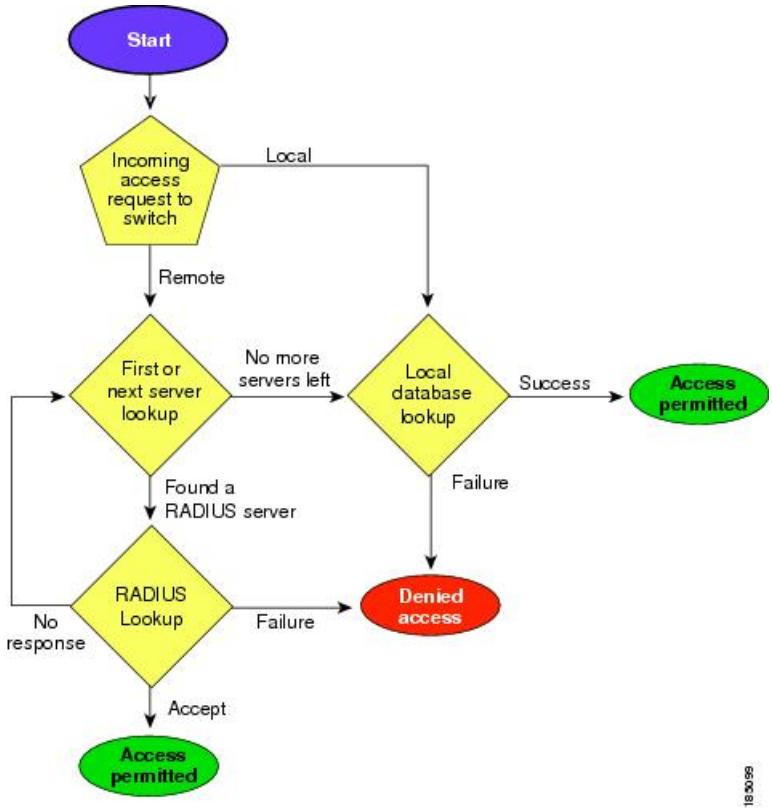
  If all configured methods fail, the local database is used for authentication.
- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

  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.

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

Configuring AAA

Configuring Console Login Authentication Methods

The authentication methods include the following:

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

The default method is local.

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

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

Procedure

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

Example

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

```
switch# configure terminal
switch(config)# aaa authentication login console group radius
switch(config)# exit
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The default method is local.

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

Procedure

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

Enabling Login Authentication Failure Messages

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login error-enable**	Enables login authentication failure messages. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.

	Command or Action	Purpose
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.

Configuring Console Authorization Commands

The authorization methods include the following:

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

The default method is local.

Before you configure console authorization commands, configure TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authorization commands console** {**group** *group-list* [**none**] | **local** | **none**}	Configures authorization for the console. The *group-list* argument consists of a space-delimited list of group name. The group name is: • *named-group* —Uses a named subset of TACACS+ servers for authorization. The **local** method uses the local database for authorization. The **none** method uses the username only. The default console authorization 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 authorization**	Displays the configuration of the console authorization commands.
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 console authorization commands:

```
switch# configure terminal
switch(config)# aaa authorization commands console group tacacs+
switch(config)# exit
switch# show aaa authorization
switch# copy running-config startup-config
```

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.

Related Topics

VSAs, on page 21

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.

	Command or Action	Purpose
		The default method is **local**, which is used when no server groups are configured or when all the configured server group do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa accounting**	Displays the configuration AAA accounting default methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Using AAA Server VSAs

VSAs

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

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

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

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

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

VSA Format

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

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

The following attributes are supported by the Cisco Nexus device:

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

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

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

```
shell:roles="roleA roleB …"
```

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

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

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

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

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

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

Secure Login Enhancements

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

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:**	Configures your Cisco NX-OS device for login parameters that help provide DoS detection.

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

Configuration Examples for Login Parameters

Setting Login Parameters Example

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

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

Showing Login Parameters Example

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

```
Switch# show login

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

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

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

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

```
Switch# show login failures

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

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

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

Configuring Login Block Per User

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable only for local users. Use this task to configure login parameters to block an user after failed login attempts.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**aaa authentication rejected** *attempts* **in** *seconds* **ban** *seconds* **Example:** `switch(config)# aaa authentication rejected 3 in 20 ban 300`	Configures login parameters to block an user. **Note** Use the **no aaa authentication rejected** command to revert to the default login parameters.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show running config** **Example:** `switch# show running config`	(Optional) Displays the login parameters.

	Command or Action	Purpose
Step 5	**show aaa local user blocked** **Example:** `switch# show aaa local user blocked`	(Optional) Displays the blocked local users.
Step 6	**clear aaa local user blocked** {**username** *user* | **all**} **Example:** `switch# clear aaa local user blocked username testuser`	(Optional) Clears the blocked local users. • **all**–Clears all the blocked local users.

Configuration Examples for Login Block Per User

Setting Parameters for Login Block Per User

The following example shows how to configure the login parameters to block a user for 300 seconds when five login attempts fail within a period of 60 seconds:

```
switch(config)# aaa authentication rejected 5 in 60 ban 300
```

Showing Login Parameters

The following example shows the login parameters configured for a switch:

```
switch# show run | i rejected
aaa authentication rejected 5 in 60 ban 300
```

Showing Blocked Local Users

The following example shows the blocked local users:

```
switch# show aaa local user blocked
Local-user                  State

testuser                    Watched (till 11:34:42 IST Feb 5 2015)
```

Clearing Blocked Local Users

The following example shows how to clear the blocked local user testuser:

```
switch# clear aaa local user blocked username testuser
```

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.

Configuring Passphrase Length

Use this task to configure the maximum and minimum passphrase length.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**userpassphrase** {{**min-length** *value* | **max-length** *value*} | **min-length** *value* **max-length** *value*} **Example:** `switch(config)# userpassphrase max-length 127`	Configures the user passphrase length. The range of minimum passphrase length values are from 8 to 127. The range of maximum passphrase length values are from 80 to 127. The default minimum passphrase length is 8 and the default maximum passphrase length is 127.
Step 3	**no userpassphrase** {**min-length** | **max-length** | **length**} **Example:** `switch(config)# no userpassphrase max-length`	Resets the passphrase length configuration to the default configuration.
Step 4	**exit** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`switch(config)# exit`	
Step 5	**show userpassphrase** {**min-length** | **max-length** | **length**} **Example:** `switch# show userpassphrase length`	Displays the maximum and minimum user passphrase length.

Configuring Passphrase Time Values

You can configure the following passphrase time values for a user:

- Lifetime – Life time of a passphrase in days. After the passphrase expires, the user is prompted to change the passphrase upon first login.
- Gracetime – Grace time of a passphrase in days. Gracetime is the number of days of inactivity after a passphrase has expired before an account is locked.
- Warntime – Warning time of the expiry of a passphrase in days. Warntime is the number of days prior to a passphrase expiring, when a user is warned that the user's passphrase is about to expire.

  The default time values are 99999 days for lifetime, 14 days for warntime, and 3 days for gracetime. The value 99999 indicates that a user's passphrase never expires by default.

Note By default, an extra configuration is added to the running configuration for every user except 'admin'. This indicates a user's passphrase time values. By default, the extra configuration displays the default passphrase time values for users.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**username** *username* **passphrase** {{**lifetime** | **warntime** | **gracetime**} *time-value* | {**lifetime** *time-value* **warntime** *time-value* **gracetime** *time-value*}} **Example:** `switch(config)# username test-user passphrase lifetime 990`	Configures passphrase time values for a user. Note that this step can be performed only by a network-admin.

	Command or Action	Purpose
Step 3	(Optional) **no username** *username* **passphrase** {**lifetime** | **warntime** | **gracetime** | **timevalues**} **Example:** `switch(config)# no username test-user passphrase lifetime`	Resets passphrase time value to default values for a user. Note that this step can be performed only by a network-admin.
Step 4	(Optional) **userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime**} *time-value* **Example:** `switch(config)# userpassphrase default-lifetime 990`	Updates default passphrase time values. Note that this step can be performed only by a network-admin.
Step 5	(Optional) **no userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime timevalue**} **Example:** `switch(config)# no userpassphrase default-lifetime`	Resets the configured default values to the initial default values. Note that this step can be performed only by a network-admin.
Step 6	(Optional) **username** *username* **expire-userpassphrase** **Example:** `switch(config)# username john expire-userpassphrase`	Sets any userpassphrase to expire immediately. When you try to log in after a passphrase expires, you are prompted to enter and create a new password after entering the old password correctly. Note that this step can be performed only by an admin.
Step 7	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 8	**show userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime** | **timevalues**} **Example:** `switch# show userpassphrase default-lifetime`	Displays the passphrase time values.
Step 9	**show username** *username* **passphrase timevalues** **Example:**	Displays the passphrase lifetime, warning time, and grace time for a specific user.

	Command or Action	Purpose
	`switch# show username john passphrase timevalues`	
Step 10	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the configured values.

Configuring Passphrase Time Values

The following example shows how to configure passphrase time values for test-user.

```
switch(config)# username test-user passphrase lifetime 365 warntime 10 gracetime 5
switch(config)# show username test-user passphrase timevalues
Last passphrase change(Y-M-D):  2016-01-28
Passphrase lifetime:            365 days after last passphrase change
Passphrase warning time starts: 10 days before passphrase lifetime
Passphrase Gracetime ends:      5 days after passphrase lifetime

switch# show running-config

!Command: show running-config
!Time: Mon Nov 30 02:32:51 2015

version 7.3(0)N1(1)
hostname switch

role name test
username admin password 5 5$0sCUUZQm$fXdGj90e9yXv1XeuY9qResKmLGKQtn8Tj6ab4s4IcVA  role
network-admin username test-user password 5
5$c9Gmvm8E$aoSQ1X7vfphlJ6WeRQl3C0Py6TlpiDjhWcF6kYi4hg6 expire 1970-01-01 role network-operator

username test-user passphrase  lifetime 365 warntime 10 gracetime 5
```

Locking User Accounts

As an admin, you can lock or unlock any user account.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] username** *username* **lock-user-account** **Example:** `switch(config)# username john lock-user-account`	Locks the specified user account. Use the **no** form of this command to unlock a user account.

	Command or Action	Purpose
Step 3	(Optional) **unlock locked-users** **Example:** `switch(config)# unlock locked-users`	Unlocks all the locked user accounts.
Step 4	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show locked-users** **Example:** `switch# show locked-users`	Displays all the locked users.

Logging Invalid Usernames

As an admin, you can ensure non-logging or logging of invalid usernames in logs during an authentication failure. By default, invalid usernames during authentication failures are not logged. Any username that does not pass authentication is considered as an invalid username and it is not logged, because when a password is entered in the username field by mistake, it can get logged. This feature can be used to mitigate the risk of logging passwords.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] aaa authentication login invalid-username-log** **Example:** `switch(config)# aaa authentication login invalid-username-log`	Enables the logging of invalid usernames during an authentication failure. Use the **no** form of this command to disable the logging of invalid usernames.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show aaa authentication login invalid-username-log** **Example:**	Displays whether logging invalid names is enabled.

	Command or Action	Purpose
	`switch# show aaa authentication login invalid-username-log`	

Changing Password

Use this task to change the password.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 To change the password, perform one of the following:

- Authenticate with the old password and then enter the new password:

  switch(config)# **change-password**

  Note By default, **password secure-mode** is enabled. So, users must use the old password for authentication before changing the password. An admin user can disable password secure-mode by using the **no password secure-mode** command. This enables users to change password without authenticating with the old password by using the **username** *username* **password** *new_password* command.

- If password secure-mode is enabled, an admin user can still use the **username** command to change password:

  switch(config)# **username admin password** *new-password* **role** *role-name*

  Note If password secure-mode is disabled, any user can use the **username** command to change the password.

Step 3 Exit to the privileged mode:

switch(config)# **exit**

Step 4 Display the status of password secure-mode:

switch# **show password secure-mode**

Changing Password

This example shows a running configuration to change the password. Replace the placeholders with relevant values for your setup.

```
config t
 change-password
 Enter old password:
 Enter new password:
```

```
Confirm new password:
exit
```

Enabling the Password Prompt for User Name

Procedure

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

Support over SHA-256 Algorithm for Verifying OS Integrity

Use the **show file bootflash:/ sha256sum** command to display the sha256sum of the file. The sample output for this command is shown below:

```
Switch# show file bootflash:/ sha256sum

abd9d40020538acc363df3d1bae7d1df16841e4903fca2c07c7898bf4f549ef5
```

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.

	Command or Action	Purpose
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.

Command	Purpose
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

Configuration Examples for AAA

The following example shows how to configure AAA:

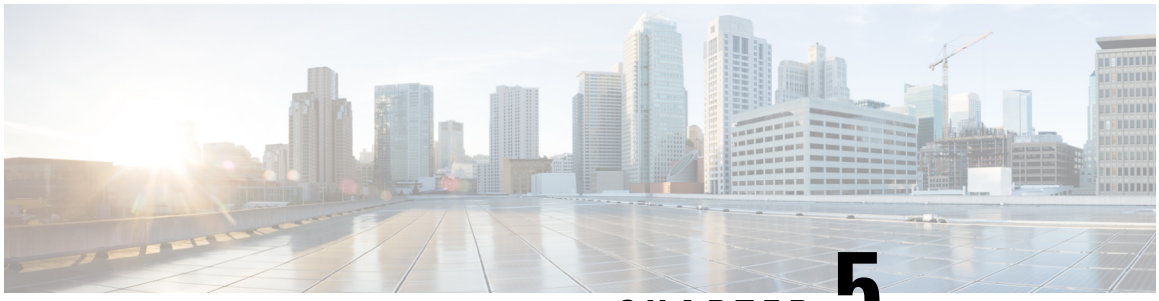
```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

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 5

Configuring RADIUS

This chapter contains the following sections:

Configuring RADIUS

Information About RADIUS

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

RADIUS Network Environments

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

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

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

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

- Networks already using RADIUS.

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

- Networks that require resource accounting.

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

- Networks that support authentication profiles.

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

Information About RADIUS Operations

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

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

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

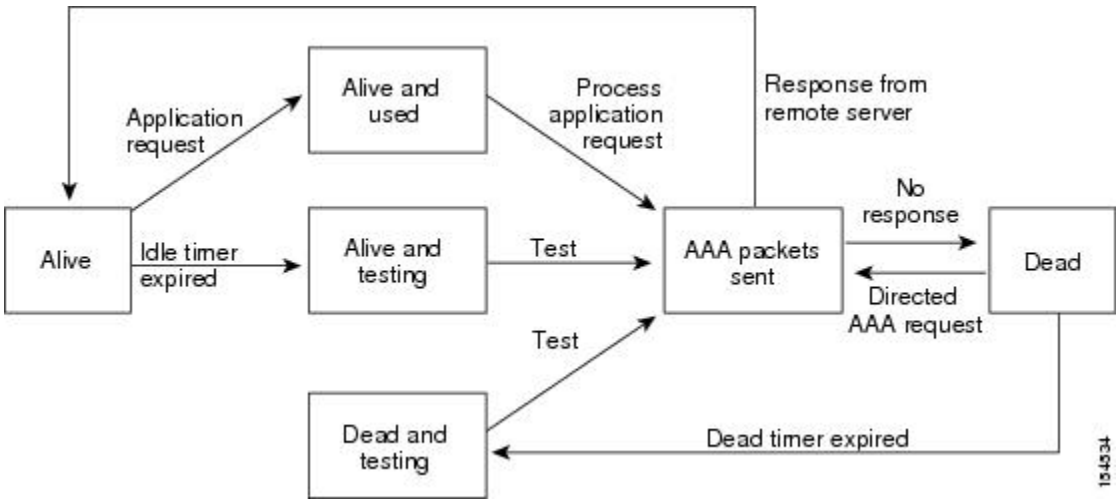
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

RADIUS Server Monitoring

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

The following figure shows the different RADIUS server states:

Figure 2: RADIUS Server States



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

Vendor-Specific Attributes

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

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

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

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

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

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

The Cisco Nexus device supports the following attributes:

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

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

Prerequisites for RADIUS

RADIUS has the following prerequisites:

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

Guidelines and Limitations for RADIUS

RADIUS has the following configuration guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco Nexus device.
- ASCII (PAP) Authentication is not supported on RADIUS servers.

Configuring RADIUS Servers

This section describes how to configure RADIUS servers.

Procedure

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

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

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

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

- Dead-time interval.
- Allow specification of a RADIUS server at login.
- Transmission retry count and timeout interval.
- Accounting and authentication attributes.

Step 5 If needed, configure periodic RADIUS server monitoring.

Configuring RADIUS Server Hosts

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

Procedure

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

Example

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

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

Configuring RADIUS Global Preshared Keys

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

Before you begin

Obtain the preshared key values for the remote RADIUS servers

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server key** [**0** | **7**] *key-value*	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters.

	Command or Action	Purpose
		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-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

Configuring RADIUS Server Preshared Keys

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

Before you begin

Obtain the preshared key values for the remote RADIUS servers.

Procedure

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

	Command or Action	Purpose
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-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS preshared keys:

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

Configuring RADIUS Server Groups

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive, alphanumeric string with a maximum 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.

	Command or Action	Purpose
Step 4	(Optional) switch (config-radius)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-radius)# **source-interface** *interface*	Assigns a source interface for a specific RADIUS server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip radius source-interface** command.
Step 6	switch(config-radius)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server group** [*group-name*]	Displays the RADIUS server group configuration.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

What to do next

Apply the RADIUS server groups to an AAA service.

Configuring the Global Source Interface for RADIUS Server Groups

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

Procedure

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

Example

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

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

Allowing Users to Specify a RADIUS Server at Login

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

Procedure

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

Example

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

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

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

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

Procedure

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

Example

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

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

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

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

Procedure

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

Example

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

```
switch# configure terminal
switch(config)# radius-server host server1 retransmit 3
switch(config)# radius-server host server1 timeout 10
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Accounting and Authentication Attributes for RADIUS Servers

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

Procedure

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

Example

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

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

Configuring Periodic RADIUS Server Monitoring

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

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

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

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

Procedure

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

Example

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

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

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

Configuring the Dead-Time Interval

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

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

Procedure

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

Example

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

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

Manually Monitoring RADIUS Servers or Groups

Procedure

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

	Command or Action	Purpose
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

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
```

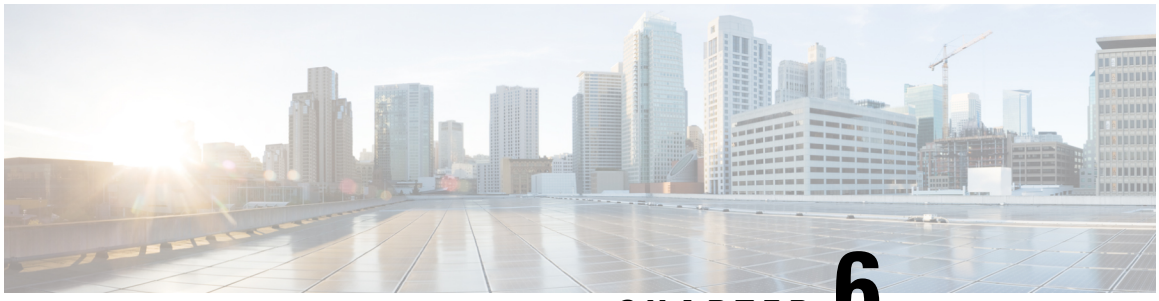
```
switch(config)# radius-server key 7 "ToIkLhPpG"
switch(config)# radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# exit
switch(config-radius)# use-vrf management
```

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



CHAPTER 6

Configuring TACACS+

This chapter contains the following sections:

About Configuring TACACS+

Information About Configuring TACACS+

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

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

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

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

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

User Login with TACACS+

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

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

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

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

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

Services include the following:

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

Default TACACS+ Server Encryption Type and Preshared Key

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

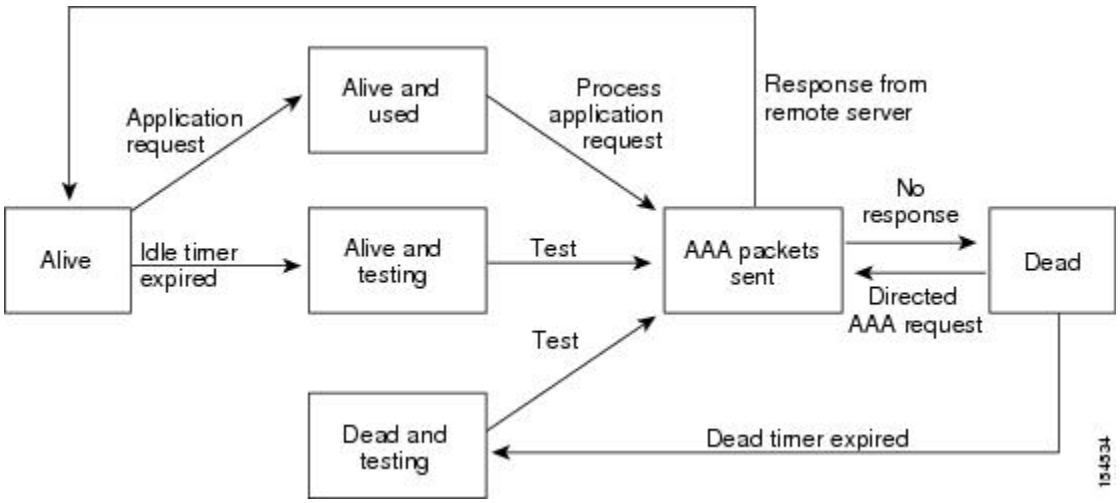
You can override the global preshared key assignment by using the **key** option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

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

The following figure shows the different TACACS+ server states:

Figure 3: TACACS+ Server States



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

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

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

Guidelines and Limitations for TACACS+

TACACS+ has the following configuration guidelines and limitations:

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

Configuring TACACS+

TACACS+ Server Configuration Process

This section describes how to configure TACACS+ servers.

Procedure

Step 1 Enable TACACS+.

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

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

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

Step 5 If needed, configure any of the following optional parameters:

- Dead-time interval
- Allow TACACS+ server specification at login
- Timeout interval
- TCP port

Step 6 If needed, configure periodic TACACS+ server monitoring.

Enabling TACACS+

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

Procedure

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

Configuring TACACS+ Server Hosts

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

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

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

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

Procedure

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

Example

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

Configuring TACACS+ Global Preshared Keys

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server key** [**0** | **7**] *key-value*	Specifies a preshared key for all TACACS+ 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 tacacs-server**	Displays the TACACS+ server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure global preshared keys:

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

Configuring TACACS+ Server Preshared Keys

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

Procedure

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

Example

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

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg
```

```
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring TACACS+ Server Groups

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

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

Before you begin

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

Procedure

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

	Command or Action	Purpose
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Configuring the Global Source Interface for TACACS+ Server Groups

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

Procedure

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

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

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

Specifying a TACACS+ Server at Login

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

Note User specified logins are only supported for Telnet sessions.

Procedure

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

Configuring the Global TACACS+ Timeout Interval

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

Procedure

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

Configuring the Timeout Interval for a Server

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

Procedure

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

Configuring TCP Ports

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

Procedure

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

Example

The following example shows how to configure TCP ports:

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

Configuring Periodic TACACS+ Server Monitoring

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

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

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

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

Procedure

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

Example

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

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

Configuring the Dead-Time Interval

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

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

Procedure

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

Configuring ASCII Authentication

You can enable ASCII authentication on the TACACS+ server.

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication login ascii-authentication** **Example:** `switch(config)# aaa authentication login ascii-authentication`	Enables ASCII authentication. The default is disabled.

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

Step 4	(Optional) **show tacacs-server** **Example:**	Displays the TACACS+ server configuration.

	Command or Action	Purpose
	`switch# show tacacs-server`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Monitoring TACACS+ Servers or Groups

Procedure

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

Example

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

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

Disabling TACACS+

You can disable TACACS+.

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

Procedure

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

Displaying TACACS+ Statistics

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

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+ configuration information, perform one of the following tasks:

Procedure

	Command or Action	Purpose
Step 1	switch# **show tacacs+** {**status** | **pending** | **pending-diff**}	Displays the TACACS+ Cisco Fabric Services distribution status and other details.
Step 2	switch# **show running-config tacacs** [**all**]	Displays the TACACS+ configuration in the running configuration.
Step 3	switch# **show startup-config tacacs**	Displays the TACACS+ configuration in the startup configuration.

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:

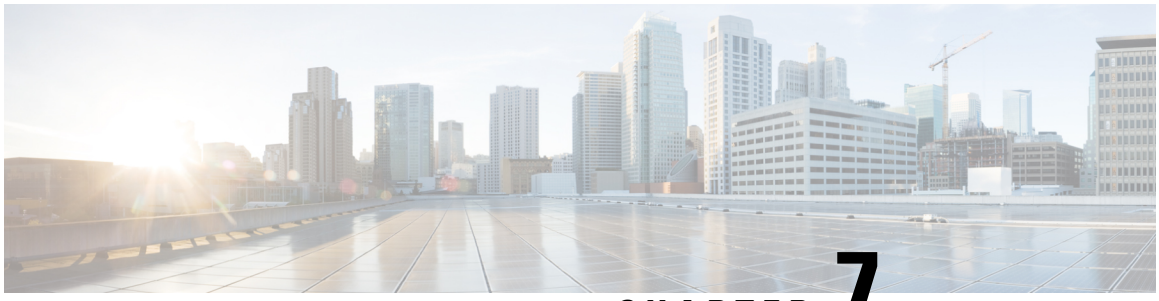
```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ikvhw10"
switch(config)# tacacs-server host 1.1.1.1
switch(config)# tacacs-server host 1.1.1.2

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 7: 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 7

Configuring SSH and Telnet

This chapter contains the following sections:

Configuring SSH and Telnet

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 Algrorithm (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 of the SSH keys, you cannot start the SSH services.

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).
- The 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 Cisco Nexus device is reloaded.

Configuring SSH

Generating SSH Server Keys

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key.

	Command or Action	Purpose
		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**	Displays the SSH server 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 generate an SSH server key:

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

Specifying the SSH Public Keys for User Accounts

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

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

Specifying the SSH Public Keys in Open SSH Format

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

Procedure

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

	Command or Action	Purpose
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
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4WlAV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

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

Specifying the SSH Public Keys in IETF SECSH Format

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

Procedure

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

Example

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

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
```

```
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

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

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

Procedure

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

Example

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

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

Starting SSH Sessions to Remote Devices

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

Procedure

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

Clearing SSH Hosts

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

Procedure

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

Disabling the SSH Server

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

Procedure

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

Deleting SSH Server Keys

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

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

Procedure

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

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

Procedure

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

Configuration Examples for SSH

The following example shows how to configure SSH:

Procedure

Step 1 Generate an SSH server key.

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

Step 2 Enable the SSH server.

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

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

Step 3 Display the SSH server key.

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

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

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

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

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

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

Step 5 Save the configuration.

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

Configuring Telnet

Enabling the Telnet Server

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

Procedure

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

Reenabling the Telnet Server

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

Procedure

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

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:

Procedure

- switch# **show ssh key** [**dsa** | **rsa**]

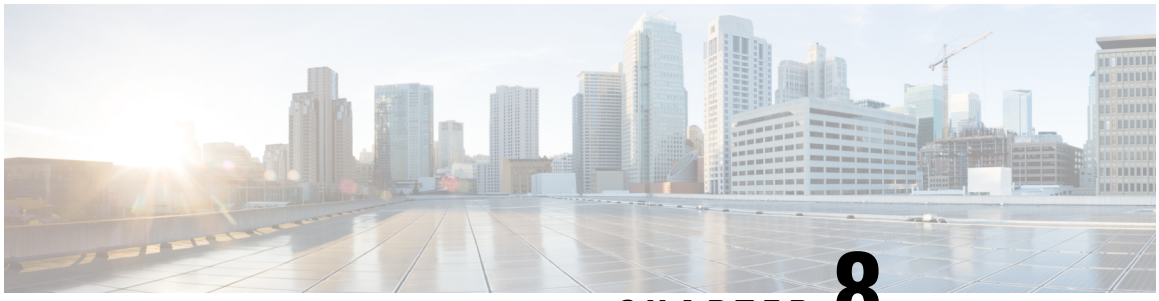
Command or Action	Purpose
switch# **show running-config security**[**all**]	Displays the SSH and user account configuration in the running configuration. The **all**keyword displays the default values for the SSH and user accounts.
switch# **show ssh server**	Displays the SSH server configuration.
switch# **show user-account**	Displays user account information

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 8: 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	Disabled



CHAPTER 8

Configuring 802.1X

This chapter contains the following sections:

Information About 802.1X

802.1X defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a Cisco NX-OS device port.

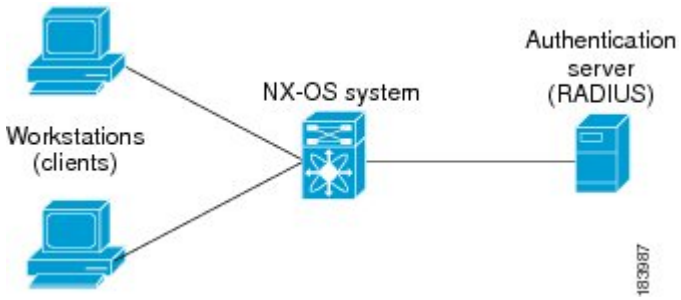
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles.

Figure 4: 802.1X Device Roles

This figure shows the device roles in 802.1X.



The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and Cisco NX-OS device services and responds to requests from the Cisco NX-OS device. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating device.

Note To resolve Windows XP network connectivity and Cisco 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL:
http://support.microsoft.com/support/kb/articles/Q303/5/97.ASP

Authentication server

The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the Cisco NX-OS device regarding whether the supplicant is authorized to access the LAN and Cisco NX-OS device services. Because the Cisco NX-OS device acts as the proxy, the authentication service is transparent to the supplicant. The Remote Authentication Dial-In User Service (RADIUS) security device with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server, version 3.0. RADIUS uses a supplicant-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

Authenticator

The authenticator controls the physical access to the network based on the authentication status of the supplicant. The authenticator acts as an intermediary (proxy) between the supplicant and the authentication server, requesting identity information from the supplicant, verifying the requested identity information with the authentication server, and relaying a response to the supplicant. The authenticator includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the authenticator receives EAPOL frames and relays them to the authentication server, the authenticator strips off the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format. This encapsulation process does not modify or examine the EAP frames, and the authentication server must support EAP within the native frame format. When the authenticator receives frames from the authentication server, the authenticator removes the server's frame header, leaving the EAP frame, which the authenticator then encapsulates for Ethernet and sends to the supplicant.

Note The Cisco NX-OS device can only be an 802.1X authenticator.

Authentication Initiation and Message Exchange

Either the authenticator (Cisco NX-OS device) or the supplicant (client) can initiate authentication. If you enable authentication on a port, the authenticator must initiate authentication when it determines that the port link state transitions from down to up. The authenticator then sends an EAP-request/identity frame to the supplicant to request its identity (typically, the authenticator sends an initial identity/request frame followed by one or more requests for authentication information). When the supplicant receives the frame, it responds with an EAP-response/identity frame.

If the supplicant does not receive an EAP-request/identity frame from the authenticator during bootup, the supplicant can initiate authentication by sending an EAPOL-start frame, which prompts the authenticator to request the supplicant's identity.

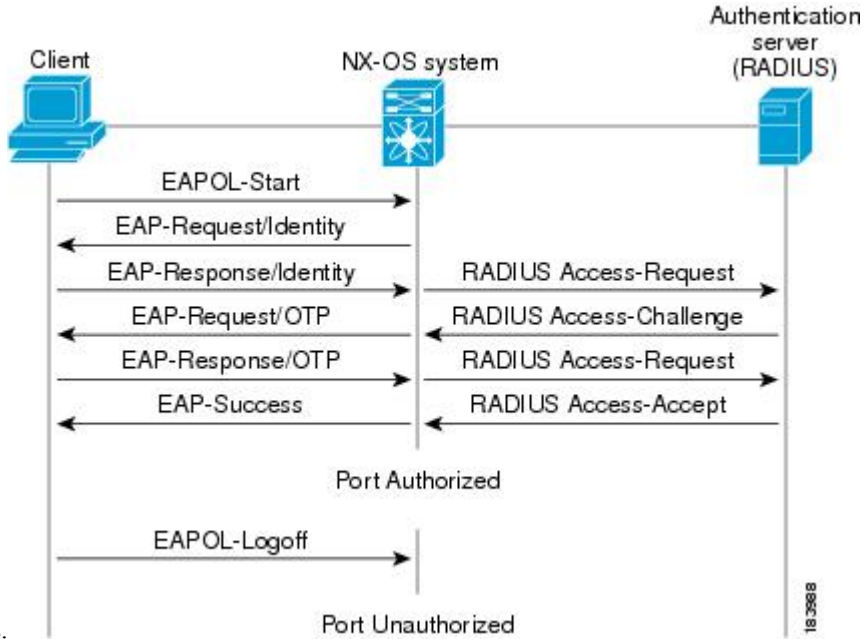
Note If 802.1X is not enabled or supported on the network access device, the Cisco NX-OS device drops any EAPOL frames from the supplicant. If the supplicant does not receive an EAP-request/identity frame after three attempts to start authentication, the supplicant transmits data as if the port is in the authorized state. A port in the authorized state means that the supplicant has been successfully authenticated.

When the supplicant supplies its identity, the authenticator begins its role as the intermediary, passing EAP frames between the supplicant and the authentication server until authentication succeeds or fails. If the authentication succeeds, the authenticator port becomes authorized.

The specific exchange of EAP frames depends on the authentication method being used.

Figure 5: Message Exchange

This figure shows a message exchange initiated by the supplicant using the One-Time-Password (OTP) authentication method with a RADIUS server. The OTP authentication device uses a secret pass-phrase to generate a sequence of one-time (single use) passwords.



The user's secret pass-phrase never crosses the network at any time such as during authentication or during pass-phrase changes.

Authenticator PAE Status for Interfaces

When you enable 802.1X on an interface, the Cisco NX-OS software creates an authenticator port access entity (PAE) instance. An authenticator PAE is a protocol entity that supports authentication on the interface. When you disable 802.1X on the interface, the Cisco NX-OS software does not automatically clear the authenticator PAE instances. You can explicitly remove the authenticator PAE from the interface and then reapply it, as needed.

Ports in Authorized and Unauthorized States

The authenticator port state determines if the supplicant is granted access to the network. The port starts in the unauthorized state. In this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a supplicant is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the supplicant to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the authenticator requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

Ports can have the following authorization states:

Force authorized
: Disables 802.1X port-based authentication and transitions to the authorized state without requiring any authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This authorization state is the default.

Force unauthorized
: Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The authenticator cannot provide authentication services to the client through the interface.

Auto
: Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received from the supplicant. The authenticator requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each supplicant that attempts to access the network is uniquely identified by the authenticator by using the supplicant's MAC address.

If the supplicant is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated supplicant are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the authenticator can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and the supplicant is not granted network access.

When a supplicant logs off, it sends an EAPOL-logoff message, which causes the authenticator port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

MAC Authentication Bypass

You can configure the Cisco NX-OS device to authorize a supplicant based on the supplicant MAC address by using the MAC authentication bypass feature. For example, you can enable this feature on interfaces configured for 802.1X that are connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the supplicant, the Cisco NX-OS device tries to authorize the client by using MAC authentication bypass.

When you enable the MAC authentication bypass feature on an interface, the Cisco NX-OS device uses the MAC address as the supplicant identity. The authentication server has a database of supplicant MAC addresses that are allowed network access. After detecting a client on the interface, the Cisco NX-OS device waits for an Ethernet packet from the client. The Cisco NX-OS device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the Cisco NX-OS device grants the client access to the network. If authorization fails, the Cisco NX-OS device assigns the port to the guest VLAN if one is configured.

If an EAPOL packet is detected on the interface during the lifetime of the link, the Cisco NX-OS device determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the Cisco NX-OS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the Cisco NX-OS device does not unauthorize the client connected to the interface. When reauthentication occurs, the Cisco NX-OS device uses 802.1X authentication as the preferred reauthentication process if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN. If reauthentication fails, the switch assigns the port to the guest VLAN, if one is configured.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*.

MAC authentication bypass interacts with the following features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.
- Port security— You can configure 802.1X authentication and port security on the same Layer 2 ports.
- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

802.1X and Port Security

You can configure port security and 802.1X on the same interfaces. Port security secures the MAC addresses that 802.1X authenticates. 802.1X processes packets before port security processes them, so when you enable both on an interface, 802.1X is already preventing inbound traffic on the interface from unknown MAC addresses.

When you enable 802.1X and port security on the same interface, port security continues to learn MAC addresses by the sticky or dynamic method, as configured. Additionally, depending on whether you enable 802.1X in single-host mode or multiple-host mode, one of the following occurs:

Single host mode

Port security learns the MAC address of the authenticated host.

Multiple host mode

Port security drops any MAC addresses learned for this interface by the dynamic method and learns the MAC address of the first host authenticated by 802.1X.

If a MAC address that 802.1X passes to port security would violate the applicable maximum number of secure MAC addresses, the device sends an authentication failure message to the host.

The device treats MAC addresses authenticated by 802.1X as though they were learned by the dynamic method, even if port security previously learned the address by the sticky or static methods. If you attempt to delete a secure MAC address that has been authenticated by 802.1X, the address remains secure.

If the MAC address of an authenticated host is secured by the sticky or static method, the device treats the address as if it were learned by the dynamic method, and you cannot delete the MAC address manually.

Port security integrates with 802.1X to reauthenticate hosts when the authenticated and secure MAC address of the host reaches its port security age limit. The device behaves differently depending upon the type of aging, as follows:

Absolute

Port security notifies 802.1X and the device attempts to reauthenticate the host. The result of reauthentication determines whether the address remains secure. If reauthentication succeeds, the device restarts the aging timer on the secure address; otherwise, the device drops the address from the list of secure addressees for the interface.

Inactivity

Port security drops the secure address from the list of secure addresses for the interface and notifies 802.1X. The device attempts to reauthenticate the host. If reauthentication succeeds, port security secures the address again.

Dynamic VLAN Assignment based on MAC-Based Authentication (MAB)

The Cisco Nexus 5000 and 6000 series switches supports dynamic VLAN assignment. After the 802.1x authentication or MAB is completed; before bringing up the port, you may want to (as part of authorization) allow the peer/host to be placed into a particular VLAN based as a result of the authentication. The RADIUS server typically indicates the desired VLAN by including tunnel attributes within the Access-Accept message. This procedure of getting the VLAN an binding it to the port constitutes to Dynamic VLAN assignment.

VLAN Assignment from RADIUS

After authentication is completed either through dot1x or MAB, the response from the RADIUS server can have dynamic VLAN information, which can be assigned to a port. This information is present in response from RADIUS server in Accept-Access message in the form of tunnel attributes. For use in VLAN assignment, the following tunnel attributes are sent:

- Tunnel-type=VLAN(13)
- Tunnel-Medium-Type=802
- Tunnel-Private-Group-ID=VLANID

All the three parameters must be received for configuring access VLAN.

Single Host and Multiple Hosts Support

The 802.1X feature can restrict traffic on a port to only one endpoint device (single-host mode) or allow traffic from multiple endpoint devices on a port (multi-host mode).

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the Cisco NX-OS device puts the port in the authorized state. When the endpoint device leaves the port, the Cisco NX-OS device put the port back into the unauthorized state. A security violation in 802.1X is defined as a detection of frames sourced from any MAC address other than the single MAC address authorized as a result of successful authentication. In this case, the interface on which this security association violation is detected (EAPOL frame from the other MAC address) will be disabled. Single host mode is applicable only for host-to-switch topology and when a single host is connected to the Layer 2 (Ethernet access port) or Layer 3 port (routed port) of the Cisco NX-OS device.

Only the first host has to be authenticated on the 802.1X port configured with multiple host mode. The port is moved to the authorized state after the successful authorization of the first host. Subsequent hosts are not required to be authorized to gain network access once the port is in the authorized state. If the port becomes unauthorized when reauthentication fails or an EAPOL logoff message is received, all attached hosts are denied access to the network. The capability of the interface to shut down upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies.

Supported Topologies

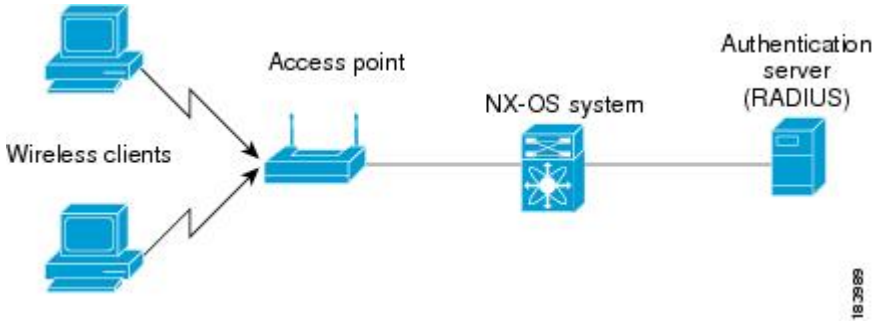
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration, only one supplicant (client) can connect to the 802.1X-enabled authenticator (Cisco NX-OS device) port. The authenticator detects the supplicant when the port link state changes to the up state. If a supplicant leaves or is replaced with another supplicant, the authenticator changes the port link state to down, and the port returns to the unauthorized state.

Figure 6: Wireless LAN Example

This figure shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one supplicant is authenticated.



When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the Cisco NX-OS device denies access to the network to all of the attached supplicants.

Licensing Requirements for 802.1X

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	802.1X requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

Prerequisites for 802.1X

802.1X Guidelines and Limitations

802.1X port-based authentication has the following configuration guidelines and limitations:

- The Cisco NX-OS software supports 802.1X authentication only on physical ports.
- The Cisco NX-OS software does not support 802.1X authentication on port channels or subinterfaces.
- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.
- The Cisco NX-OS software supports 802.1X authentication only on Ethernet interfaces that are in a port channel, a trunk, or an access port.
- The Cisco NX-OS software does not support single host mode on trunk interfaces or member interfaces in a port channel.
- The Cisco NX-OS software does not support MAC address authentication bypass on trunk interfaces.
- The Cisco NX-OS software does not support MAC address authentication bypass on a port channel.

- The Cisco NX-OS software does not support Dot1X on vPC ports and MCT.
- The Cisco NX-OS software does not support the following 802.1X protocol enhancements:
  - One-to-many logical VLAN name to ID mapping
  - Web authorization
  - Dynamic domain bridge assignment
  - IP telephony

Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

Table 9: Default 802.1X Parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (**force-authorized**) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet timeout period	60 seconds (number of seconds that the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant)
Retransmission timeout period	30 seconds (number of seconds that the Cisco NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)
Maximum retransmission number	2 times (number of times that the Cisco NX-OS device will send an EAP-request/identity frame before restarting the authentication process)
Host mode	Single host
Supplicant timeout period	30 seconds (when relaying a request from the authentication server to the supplicant, the amount of time that the Cisco NX-OS device waits for a response before retransmitting the request to the supplicant)

Parameters	Default
Authentication server timeout period	30 seconds (when relaying a response from the supplicant to the authentication server, the amount of time that the Cisco NX-OS device waits for a reply before retransmitting the response to the server)

Configuring 802.1X

This section describes how to configure the 802.1X feature.

Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

Procedure

Step 1 Enable the 802.1X feature.
Step 2 Configure the connection to the remote RADIUS server.
Step 3 Enable 802.1X feature on the Ethernet interfaces.

Enabling the 802.1X Feature

You must enable the 802.1X feature on the Cisco NX-OS device before authenticating any supplicant devices.

Procedure

	Command or Action	Purpose

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

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature. The default is disabled.

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

	Command or Action	Purpose
Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Authentication Methods for 802.1X

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Cisco NX-OS device can perform 802.1X authentication.

Before you begin

Obtain the names or addresses for the remote RADIUS server groups.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication dot1x default group** *group-list* **Example:** `switch(config)# aaa authentication dot1x default group rad2`	Specifies the RADIUS server groups to use for 802.1X authentication. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses the global pool of RADIUS servers for authentication.

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

Step 4	(Optional) **show radius-server** **Example:**	Displays the RADIUS server configuration.

	Command or Action	Purpose
	`switch# show radius-server`	
Step 5	(Optional) **show radius-server group** [*group-name*] **Example:** `switch# show radius-server group rad2`	Displays the RADIUS server group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Controlling 802.1X Authentication on an Interface

You can control the 802.1X authentication performed on an interface. An interface can have the following 802.1X authentication states:

Auto
Enables 802.1X authentication on the interface.

Force-authorized
Disables 802.1X authentication on the interface and allows all traffic on the interface without authentication. This state is the default.

Force-unauthorized
Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot* / *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x port-control** {**auto** | **force-authorized** | **forced-unauthorized**} **Example:** `switch(config-if)# dot1x port-control auto`	Changes the 802.1X authentication state on the interface. The default is force-authorized.

	Command or Action	Purpose

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

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **show dot1x interface ethernet** *slot* **/** *port* **Example:** `switch# show dot1x interface ethernet 2/1`	Displays 802.1X feature status and configuration information for an interface.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating or Removing an Authenticator PAE on an Interface

You can create or remove the 802.1X authenticator port access entity (PAE) instance on an interface.

Note By default, the Cisco NX-OS software creates the authenticator PAE instance on the interface when you enable 802.1X on an interface.

Before you begin

Enable the 802.1X feature.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **show dot1x interface ethernet** *slot*/*port* **Example:** `switch# show do1x interface ethernet 2/1`	Displays the 802.1X configuration on the interface.

	Command or Action	Purpose

Step 3	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 4	[**no**] **dot1x pae authenticator** **Example:** `switch(config-if)# dot1x pae authenticator`	Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface. **Note** If an authenticator PAE already exists on the interface the **dot1x pae authentication** command does not change the configuration on the interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Periodic Reauthentication for an Interface

You can enable periodic 802.1X reauthentication on an interface and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication defaults to the global value.

Note During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**dot1x re-authentication** **Example:** `switch(config-if)# dot1x re-authentication`	Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.
Step 4	(Optional) **dot1x timeout re-authperiod** *seconds* **Example:** `switch(config-if)# dot1x timeout re-authperiod 3300`	Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535. **Note** This command affects the behavior of the Cisco NX-OS device only if you enable periodic reauthentication on the interface.

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

Step 6	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Reauthenticating Supplicants

You can manually reauthenticate the supplicants for the entire Cisco NX-OS device or for an interface.

Note During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**dot1x re-authenticate** [**interface** *slot*/*port*] **Example:**	Reauthenticates the supplicants on the Cisco NX-OS device or on an interface.

	Command or Action	Purpose
	`switch# dot1x re-authenticate interface 2/1`	

Manually Initializing 802.1X Authentication

You can manually initialize the authentication for all supplicants on a Cisco NX-OS device or for a specific interface.

Note Initializing the authentication clears any existing authentication status before starting the authentication process for the client.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**dot1x initialize** [**interface ethernet** *slot*/*port*] **Example:** `switch# dot1x initialize interface ethernet 2/1`	Initializes 802.1X authentication on the Cisco NX-OS device or on a specified interface.

Changing 802.1X Authentication Timers for an Interface

You can change the following 802.1X authentication timers on the Cisco NX-OS device interfaces:

Quiet-period timer
: When the Cisco NX-OS device cannot authenticate the supplicant, the switch remains idle for a set period of time and then tries again. The quiet-period timer value determines the idle period. An authentication failure might occur because the supplicant provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default. The default is the value of the global quiet period timer. The range is from 1 to 65535 seconds.

Rate-limit timer
: The rate-limit period throttles EAPOL-Start packets from supplicants that are sending too many EAPOL-Start packets. The authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated for the rate-limit period duration. The default value is 0 seconds and the authenticator processes all EAPOL-Start packets. The range is from 1 to 65535 seconds.

Switch-to-authentication-server retransmission timer for Layer 4 packets
: The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Cisco NX-OS device waits a set period of time and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP response frames
: The supplicant responds to the EAP-request/identity frame from the Cisco NX-OS device with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set

period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP request frames

The supplicant notifies the Cisco NX-OS device it that received the EAP request frame. If the authenticator does not receive this notification, it waits a set period of time and then retransmits the frame. The default is the value of the global retransmission period timer. The range is from 1 to 65535 seconds.

Note You should change the default values only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain supplicants and authentication servers.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	(Optional) **dot1x timeout quiet-period** *seconds* **Example:** `switch(config-if)# dot1x timeout quiet-period 25`	Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 4	(Optional) **dot1x timeout ratelimit-period** *seconds* **Example:** `switch(config-if)# dot1x timeout ratelimit-period 10`	Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.
Step 5	(Optional) **dot1x timeout server-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout server-timeout 60`	Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.

	Command or Action	Purpose
Step 6	(Optional) **dot1x timeout supp-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout supp-timeout 20`	Sets the number of seconds that the Cisco NX-OS device waits for the supplicant to respond to an EAP request frame before the Cisco NX-OS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 7	(Optional) **dot1x timeout tx-period** *seconds* **Example:** `switch(config-if)# dot1x timeout tx-period 40`	Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

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

Step 9	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays the 802.1X configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Single Host or Multiple Hosts Mode

You can enable single host or multiple hosts mode on an interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port* **Example:**	Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
	`switch(config)# interface ethernet 2/1` `switch(config-if)`	
Step 3	**dot1x host-mode** {**multi-host** | **single-host**} **Example:** `switch(config-if)# dot1x host-mode multi-host`	Configures the host mode. The default is single-host. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

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

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling MAC Authentication Bypass

You can enable MAC authentication bypass on an interface that has no supplicant connected.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**dot1x mac-auth-bypass** [**eap**] **Example:** `switch(config-if)# dot1x mac-auth-bypass`	Enables MAC authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the Cisco NX-OS device to use EAP for authorization.

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

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling 802.1X Authentication on the Cisco NX-OS Device

You can disable 802.1X authentication on the Cisco NX-OS device. By default, the Cisco NX-OS software enables 802.1X authentication after you enable the 802.1X feature. However, when you disable the 802.1X feature, the configuration is removed from the Cisco NX-OS device. The Cisco NX-OS software allows you to disable 802.1X authentication without losing the 802.1X configuration.

Note When you disable 802.1X authentication, the port mode for all interfaces defaults to force-authorized regardless of the configured port mode. When you reenable 802.1X authentication, the Cisco NX-OS software restores the configured port mode on the interfaces.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**no dot1x system-auth-control** **Example:**	Disables 802.1X authentication on the Cisco NX-OS device. The default is enabled.

	Command or Action	Purpose
	`switch(config)# no dot1x system-auth-control`	**Note** Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Cisco NX-OS device.

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

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling the 802.1X Feature

You can disable the 802.1X feature on the Cisco NX-OS device.

When you disable 802.1X, all related configurations are automatically discarded. The Cisco NX-OS software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Cisco NX-OS System Management Configuration Guide* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**no feature dot1x** **Example:** `switch(config)# no feature dot1x`	Disables 802.1X. **Caution** Disabling the 802.1X feature removes all 802.1X configuration.
Step 3	**exit** **Example:**	Exits configuration mode.

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

Setting the Maximum Authenticator-to-Supplicant Frame Retransmission Retry Count for an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits authentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x max-req** *count* **Example:** `switch(config-if)# dot1x max-req 3`	Changes the maximum authorization request retry count. The default is 2 times and the range is from 1 to 10. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

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

	Command or Action	Purpose
Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling RADIUS Accounting for 802.1X Authentication

You can enable RADIUS accounting for the 802.1X authentication activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**dot1x radius-accounting** **Example:** `switch(config)# dot1x radius-accounting`	Enables RADIUS accounting for 802.1X. The default is disabled.

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

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Accounting Methods for 802.1X

You can enable AAA accounting methods for the 802.1X feature.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting dot1x default group** *group-list*	Configures AAA accounting for 802.1X. The default is disabled. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—For all configured RADIUS servers. • *named-group*—Any configured RADIUS server group name.
Step 3	**exit**	Exits configuration mode.
Step 4	(Optional) **show aaa accounting**	Displays the AAA accounting configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the 802.1x feature:

```
switch# configure terminal
switch(config)# aaa accounting dot1x default group radius
switch(config)# exit
switch# show aaa accounting
switch# copy running-config startup-config
```

Setting the Maximum Reauthentication Retry Count on an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits reauthentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x max-reauth-req** *retry-count* **Example:** `switch(config-if)# dot1x max-reauth-req 3`	Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10.

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

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Guest VLAN

If MAB is configured, and if there is an authentication failure due to MAB, then the guest VLAN (if available), will be assigned as access VLAN.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface ethernet** *slot* / *port* **Example:**	Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
	`switch(config)# interface ethernet 2/1`	
Step 3	**dot1x guest-vlan** *guest-vlan* **Example:** `switch(config-if)# dot1x guest-vlan 5`	Specifies the guest VLAN to be assigned.
Step 4	**exit** **Example:** `switch(config-if)# exit`	Returns to privileged EXEC mode.

Verifying VLAN Assignment

```
Switch(config-if)# show interface ethernet 1/1 br

--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch
#
--------------------------------------------------------------------------------
Eth1/1        501     eth  access up      none                       1000(D) --


Switch(config-if)# show dot1x interface ethernet 1/1

Dot1x Info for Ethernet1/1
----------------------------------
                     PAE = AUTHENTICATOR
             PortControl = AUTO
                HostMode = SINGLE HOST
         ReAuthentication = Disabled
              QuietPeriod = 60
           ServerTimeout = 30
             SuppTimeout = 30
            ReAuthPeriod = 3600 (Locally configured)
               ReAuthMax = 1
                  MaxReq = 2
                 TxPeriod = 1
          RateLimitPeriod = 0
          Mac-Auth-Bypass = Enabled
                Auth_vlan = 501
```

Verifying the 802.1X Configuration

To display 802.1X information, perform one of the following tasks:

Command	Purpose
show dot1x	Displays the 802.1X feature status.
show dot1x all [**details** | **statistics** | **summary**]	Displays all 802.1X feature status and configuration information.

Command	Purpose
show dot1x interface ethernet *slot/port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an Ethernet interface.
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Cisco NX-OS Security Command Reference* for your platform.

Monitoring 802.1X

You can display the statistics that the Cisco NX-OS device maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show dot1x** {**all** | **interface ethernet** *slot/port*} **statistics** **Example:** `switch# show dot1x all statistics`	Displays the 802.1X statistics.

Configuration Example for 802.1X

The following example shows how to configure 802.1X for an access port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
```

The following example shows how to configure 802.1X for a trunk port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
dot1x host-mode multi-host
```

Note Repeat the **dot1x pae authenticator** and **dot1x port-control auto** commands for all interfaces that require 802.1X authentication.

Additional References for 802.1X

This section includes additional information related to implementing 802.1X.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	
VRF configuration	

Standards

Standards	Title
IEEE Std 802.1X- 2004 (Revision of IEEE Std 802.1X-2001)	*802.1X IEEE Standard for Local and Metropolitan Area Networks Port-Based Network Access Control*
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 3580	*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*

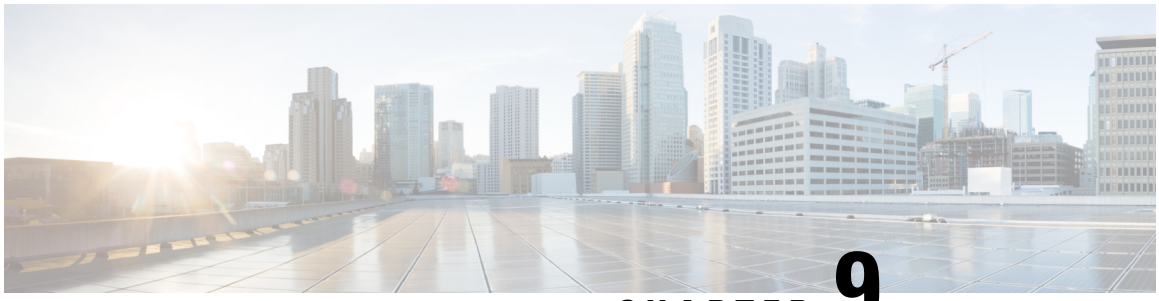
MIBs

MIBs	MIBs Link
• IEEE8021-PAE-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for 802.1X

Table 10: Feature History for 802.1X

Feature Name	Release	Feature Information
802.1X	6.0(2)N1(2)	This feature was introduced.



CHAPTER 9

Configuring Cisco TrustSec

This chapter describes how to configure Cisco TrustSec on Cisco NX-OS devices.

This chapter includes the following sections:

Information About Cisco TrustSec

This section provides information about Cisco TrustSec.

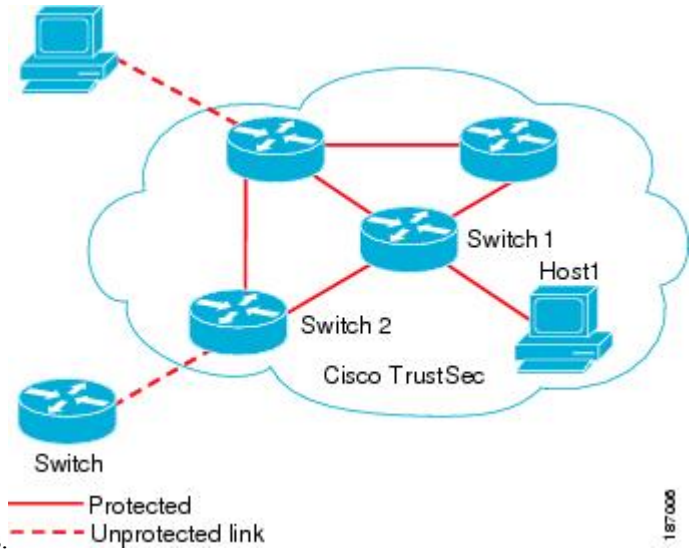
Cisco TrustSec Architecture

The Cisco TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Cisco TrustSec also uses the device information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Cisco TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Note Ingress refers to entering the first Cisco TrustSec-capable device encountered by a packet on its path to the destination and egress refers to leaving the last Cisco TrustSec-capable device on the path.

Figure 7: Cisco TrustSec Network Cloud Example

This figure shows an example of a Cisco TrustSec cloud. In this example, several networking devices and an endpoint device are inside the Cisco TrustSec cloud. One endpoint device and one networking device are outside the cloud because they are not Cisco TrustSec-capable devices.



The Cisco TrustSec architecture consists of the following major components:

Authentication
Verifies the identity of each device before allowing them to join the Cisco TrustSec network.

Authorization
Decides the level of access to the Cisco TrustSec network resources for a device based on the authenticated identity of the device.

Access control
Applies access policies on a per-packet basis using the source tags on each packet.

A Cisco TrustSec network has the following entities:

Authenticators (AT)
Devices that are already part of a Cisco TrustSec network.

Authorization server (AS)
Servers that may provide authentication information, authorization information, or both.

When the link first comes up, authorization occurs in which each side of the link obtains policies, such as SGT and ACLs, that apply to the link.

Authentication

Cisco TrustSec authenticates a device before allowing it to join the network. Cisco TrustSec uses 802.1X authentication with Extensible Authentication Protocol Flexible Authentication through Secure Tunnel (EAP-FAST) as the Extensible Authentication Protocol (EAP) method to perform the authentication.

Device Identities

Cisco TrustSec does not use IP addresses or MAC addresses as device identities. Instead, assign a name (device ID) to each Cisco TrustSec-capable Cisco NX-OS device to identify it uniquely in the Cisco TrustSec network. This device ID is used for the following:

- Looking up authorization policy
- Looking up passwords in the databases during authentication

Device Credentials

Cisco TrustSec supports password-based credentials. The authentication servers may use self-signed certificates instead. Cisco TrustSec authenticates the supplicants through passwords and uses MSCHAPv2 to provide mutual authentication even if the authentication server certificate is not verifiable.

The authentication server uses these credentials to mutually authenticate the supplicant during the EAP-FAST phase 0 (provisioning) exchange, where a PAC is provisioned in the supplicant. Cisco TrustSec does not perform the EAP-FAST phase 0 exchange again until the PAC expires and only performs EAP-FAST phase 1 and phase 2 exchanges for future link bringups. The EAP-FAST phase 1 exchange uses the PAC to mutually authenticate the authentication server and the supplicant. Cisco TrustSec uses the device credentials only during the PAC provisioning (or reprovisioning) steps.

The authentication server uses a temporarily configured password to authenticate the supplicant when the supplicant first joins the Cisco TrustSec network. When the supplicant first joins the Cisco TrustSec network, the authentication server authenticates the supplicant using a manufacturing certificate and then generates a strong password and pushes it to the supplicant with the PAC. The authentication server also keeps the new password in its database. The authentication server and the supplicant use this password for mutual authentication in all future EAP-FAST phase 0 exchanges.

User Credentials

Cisco TrustSec does not require a specific type of user credentials for endpoint devices. You can choose any type of authentication method for the user (for example, MSCHAPv2, LEAP, generic token card (GTC), or OTP) and use the corresponding credentials. Cisco TrustSec performs user authentication inside the EAP-FAST tunnel as part of the EAP-FAST phase 2 exchange.

SGACLs and SGTs

In security group access lists (SGACLs), you can control the operations that users can perform based on assigned security groups. The grouping of permissions into a role simplifies the management of the security policy. As you add users to a Cisco NX-OS device, you simply assign one or more security groups and they immediately receive the appropriate permissions. You can modify security groups to introduce new privileges or restrict current permissions.

Cisco TrustSec assigns a unique 16-bit tag, called the security group tag (SGT), to a security group. The number of SGTs in a Cisco NX-OS device is limited to the number of authenticated network entities. The SGT is a single label that indicates the privileges of the source within the entire enterprise. Its scope is global within a Cisco TrustSec network.

The management server derives the SGTs based on the security policy configuration. You do not have to configure them manually.

Once authenticated, Cisco TrustSec tags any packet that originates from a device with the SGT that represents the security group to which the device is assigned. The packet carries this SGT throughout the network within the Cisco TrustSec header. Because this tag represents the group of the source, the tag is referred to as the source SGT. At the egress edge of the network, Cisco TrustSec determines the group that is assigned to the packet destination device and applies the access control policy.

Cisco TrustSec defines access control policies between the security groups. By assigning devices within the network to security groups and applying access control between and within the security groups, Cisco TrustSec essentially achieves access control within the network.

Figure 8: SGACL Policy Example

This figure shows an example of an SGACL policy.

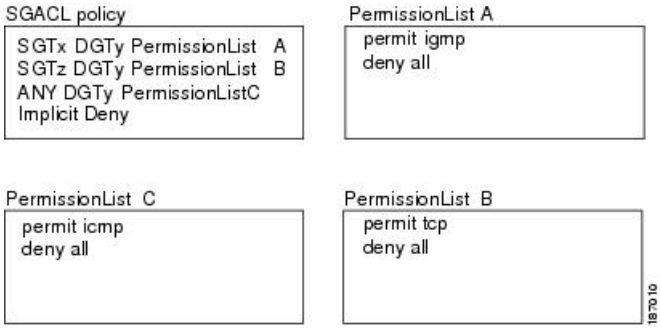
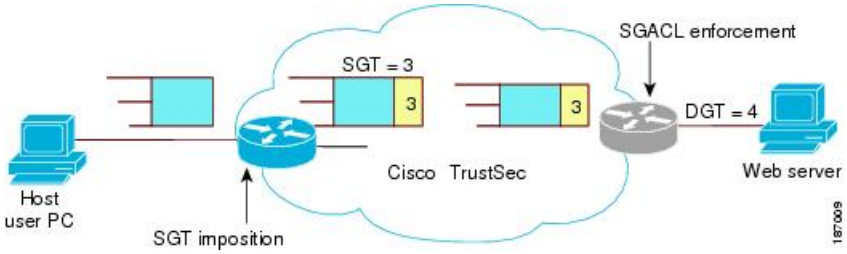


Figure 9: SGT and SGACL in Cisco TrustSec Network

This figure shows how the SGT assignment and the SGACL enforcement operate in a Cisco TrustSec network.



The Cisco NX-OS device defines the Cisco TrustSec access control policy for a group of devices as opposed to IP addresses in traditional ACLs. With such a decoupling, the network devices are free to move throughout the network and change IP addresses. Entire network topologies can change. As long as the roles and the permissions remain the same, changes to the network do not change the security policy. This feature greatly reduces the size of ACLs and simplifies their maintenance.

In traditional IP networks, the number of access control entries (ACEs) configured is determined as follows:

Number of ACEs = (number of sources specified) X (number of destinations specified) X (number of permissions specified)

Cisco TrustSec uses the following formula:

Number of ACEs = number of permissions specified

For information about SGACL policy enforcement with SGT caching, see SGACL Policy Enforcement With Cisco TrustSec SGT Caching.

Determining the Source Security Group

A network device at the ingress of the Cisco TrustSec network cloud needs to determine the SGT of the packet entering the Cisco TrustSec network cloud so that it can tag the packet with that SGT when it forwards it into the Cisco TrustSec network cloud. The egress network device needs to determine the SGT of the packet so that it can apply the SGACLs.

The network device can determine the SGT for a packet using one of the following methods:

- Obtain the source SGT during policy acquisition—After the Cisco TrustSec authentication phase, a network device acquires a policy from an authentication server. The authentication server indicates whether the peer device is trusted or not. If a peer device is not trusted, the authentication server can also provide an SGT to apply to all packets coming from the peer device.
- Obtain the source SGT field from the Cisco TrustSec header—If a packet comes from a trusted peer device, the Cisco TrustSec header carries the correct SGT field if the network device is not the first network device in the Cisco TrustSec network cloud for the packet.
- Look up the source SGT based on the source IP address—In some cases, you can manually configure the policy to decide the SGT of a packet based on the source IP address. The SGT Exchange Protocol (SXP) can also populate the IP-address-to-SGT mapping table.

Determining the Destination Security Group

The egress network device in a Cisco TrustSec network cloud determines the destination group for applying the SGACL. In some cases, ingress devices or other nonegress devices might have destination group information available. In those cases, SGACLs might be applied in these devices rather than in egress devices.

Cisco TrustSec determines the destination group for the packet in the following ways:

- Destination SGT of the egress port obtained during the policy acquisition
- Destination SGT lookup based on the destination IP address

Do not configure the destination SGT to enforce Cisco TrustSec on egress broadcast, multicast, and unknown unicast traffic on Fabric Extender (FEX) or vEthernet ports. Instead, set the DST to zero (unknown). The following is an example of the correct configuration:

```
cts role-based access-list acl-on-fex-egress
     deny udp
     deny ip
cts role-based sgt 9 dst 0 access-list acl-on-fex-egress
```

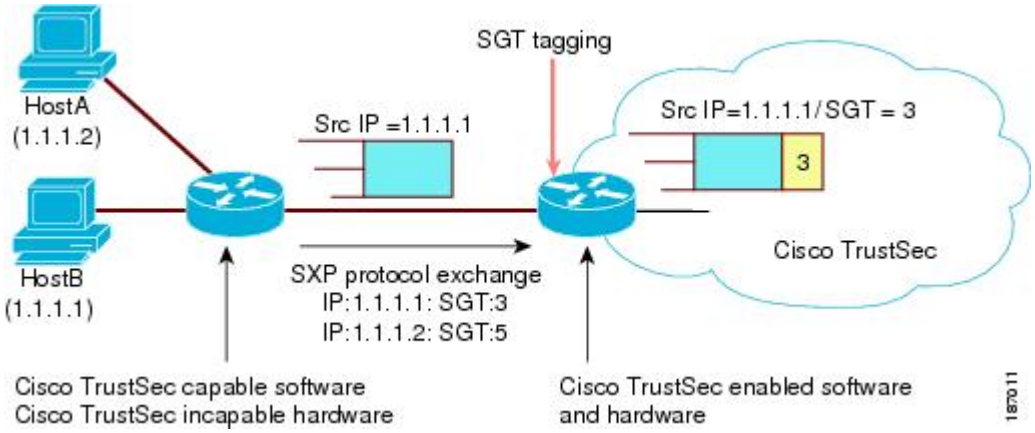
SXP for SGT Propagation Across Legacy Access Networks

The Cisco NX-OS device hardware in the access layer supports Cisco TrustSec. Without the Cisco TrustSec hardware, the Cisco TrustSec software cannot tag the packets with SGTs. You can use SXP to propagate the SGTs across network devices that do not have hardware support for Cisco TrustSec.

SXP operates between access layer devices and distribution layer devices. The access layer devices use SXP to pass the IP addresses of the Cisco TrustSec-authenticated devices with their SGTs to the distribution switches. Distribution devices with both Cisco TrustSec-enabled software and hardware can use this information to tag packets appropriately and enforce SGACL policies.

Figure 10: Using SXP to Propagate SGT Information

This figure shows how to use SXP to propagate SGT information in a legacy network.



Tagging packets with SGTs requires hardware support. You might have devices in your network that cannot tag packets with SGTs. To allow these devices to send IP address-to-SGT mappings to a device that has Cisco TrustSec-capable hardware, you must manually set up the SXP connections. Manually setting up an SXP connection requires the following:

- If you require SXP data integrity and authentication, you must configure the same SXP password on both of the peer devices. You can configure the SXP password either explicitly for each peer connection or globally for the device. The SXP password is not required.
- You must configure each peer on the SXP connection as either an SXP speaker or an SXP listener. The speaker device distributes the SXP information to the listener device.
- You can specify a source IP address to use for each peer relationship or you can configure a default source IP address for peer connections where you have not configured a specific source IP address.

Environment Data Download

The Cisco TrustSec environment data is a collection of information or policies that assists a device to function as a Cisco TrustSec node. The device acquires the environment data from the authentication server when the device first joins a Cisco TrustSec network cloud, although you might also manually configure some of the data on a device. For example, you must configure the seed Cisco TrustSec device with the authentication server information, which can later be augmented by the server list that the device acquires from the authentication server.

Note If you have manually configured the Cisco TrustSec device ID, but not using the AAA server for a Cisco TrustSec deployment, you should remove the Cisco TrustSec device ID by using the **no cts device-id** command. Otherwise, the following false syslog error is generated:

```
ENVIRONMENT_DATA_DOWNLOAD_FAILURE:  Environment data download failed from AAA
```

The **no cts device-id** command is supported from Cisco NX-OS Release 7.2. If you are using Cisco NX-OS Release 6.2.6 or a later release, you can disable only by disabling Cisco TrustSec and reapplying Cisco TrustSec configurations without the **cts device-id** configuration.

The device must refresh the Cisco TrustSec environment data before it expires. The device can also cache the data and reuse it after a reboot if the data has not expired.

The device uses RADIUS to acquire the following environment data from the authentication server:

Server lists
List of servers that the client can use for future RADIUS requests (for both authentication and authorization)

Device SGT
Security group to which the device itself belongs

Expiry timeout
Interval that controls how often the Cisco TrustSec device should refresh its environment data

Licensing Requirements for Cisco TrustSec

The following table shows the licensing requirements for this feature:

Table 11: Licensing Requirements for Cisco TrustSec

Product	License Requirement
Cisco NX-OS	Beginning with Cisco NX-OS Release 6.1, Cisco TrustSec requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For releases earlier than Cisco NX-OS 6.1, Cisco TrustSec requires an Advanced Services license. Cisco TrustSec licensing does not have a grace period. You must obtain and install an Advanced Services license before you can use Cisco TrustSec. **Note** For an explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the Cisco NX-OS Licensing Guide.

Prerequisites for Cisco TrustSec

Cisco TrustSec has the following prerequisites:

- You must install the Advanced Services license if your device is running a release earlier than Cisco NX-OS Release 6.1.
- You must enable the 802.1X feature before you enable the Cisco TrustSec feature. Although none of the 802.1X interface level features are available, 802.1X is required for the device to authenticate with RADIUS.

Guidelines and Limitations for Cisco TrustSec

Cisco TrustSec has the following guidelines and limitations:

- Cisco TrustSec uses RADIUS for authentication.

- AAA authentication and authorization for Cisco TrustSec is only supported by the Cisco Secure Access Control Server (ACS).
- Cisco TrustSec supports IPv4 addressing only.
- SXP cannot use the management (mgmt 0) interface.
- You cannot enable Cisco TrustSec on interfaces in half-duplex mode.
- Clearing policies does not take affect immediately; it requires a flap to occur. In addition, the way policies are cleared depends on whether the SGT is static or dynamic. For a static SGT, the SGT is reset to 0 after the flap occurs. For dynamic SGT, the SGT is downloaded again from the RADIUS server after the flap occurs.
- Cisco TrustSec supports management switch virtual interfaces (SVIs), not routed SVIs.
- The 802.1X feature must be enabled before you enable the Cisco TrustSec feature. However, none of the 802.1X interface level features are available. The 802.1X feature is only used for the device to authenticate with RADIUS.
- RBACL is only implemented on bridged Ethernet traffic and cannot be enabled on a routing VLAN or routing interface.
- The determination of whether a peer is trusted or not and its capability to propagate SGTs on egress are made at the physical interface level.
- Cisco TrustSec interface configurations on port channel members must be exactly the same. If a port channel member is inconsistent with the other port channel members, it will be error disabled.
- In a vPC domain, use the configuration synchronization mode (config-sync) to create switch profiles to ensure that the Cisco TrustSec configuration is synchronized between peers. If you configure the same vPC differently on two peer switches, traffic is treated differently.
- The maximum number of RBACL TCAM entries is 128, with 4 entries used by default, and the remaining 124 entries user-configurable.
- Cisco TrustSec is not supported on Layer 3 interfaces or Virtual Routing and Forwarding (VRF) interfaces.
- The **cts-manual**, **cts trusted mode**, and **no-propagate sgt** configurations must be consistent among all FEX ports or vEthernet ports on the same fabric port. If these configurations are inconsistent, the interfaces are err-disabled.
- The **cts-manual**, **sgt value**, **cts trusted mode**, and **no-propagate sgt** configurations must be consistent among all port channel members on the same port channel. If these configurations are inconsistent, the interfaces are err-disabled.

Default Settings for Cisco TrustSec Parameters

This table lists the default settings for Cisco TrustSec parameters.

Table 12: Default Cisco TrustSec Parameters Settings

Parameter	Default
Cisco TrustSec	Disabled

Parameter	Default
SXP	Disabled
SXP default password	None
SXP reconcile period	120 seconds (2 minutes)
SXP retry period	60 seconds (1 minute)
Caching	Disabled

Configuring Cisco TrustSec

This section provides information about the configuration tasks for Cisco TrustSec.

Enabling the Cisco TrustSec SGT Feature

You must enable both the 802.1X feature and the Cisco TrustSec feature on the Cisco NX-OS device before you can configure Cisco TrustSec.

Note You cannot disable the 802.1X feature after you enable the Cisco TrustSec feature.

Before you begin

Ensure that you have installed the Advanced Services license, if your device is running a release earlier than Cisco NX-OS Release 6.1.

Procedure

	Command or Action	Purpose

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

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature.
Step 3	**feature cts** **Example:** `switch(config)# feature cts`	Enables the Cisco TrustSec feature.

	Command or Action	Purpose

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

Step 5	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Cisco TrustSec configuration.
Step 6	(Optional) **show feature** **Example:** `switch# show feature`	Displays the enabled status for features.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Cisco TrustSec Device Credentials

You must configure unique Cisco TrustSec credentials on each Cisco TrustSec-enabled Cisco NX-OS device in your network. Cisco TrustSec uses the password in the credentials for device authentication.

Note You must also configure the Cisco TrustSec credentials for the Cisco NX-OS device on the Cisco Secure ACS. See the documentation at:

http://www.cisco.com/c/en/us/support/security/secure-access-control-system/products-installation-and-configuration-guides-list.html

Before you begin

Ensure that you have enabled Cisco TrustSec.

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	**cts device-id** *name* **password** *password* **Example:** `switch(config)# cts device-id MyDevice1 password CiscO321`	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive. **Note** To remove the configuration of device ID and the password, use the **no** form of the command.

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

Step 4	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Cisco TrustSec configuration.
Step 5	(Optional) **show cts environment** **Example:** `switch# show cts environment`	Displays the Cisco TrustSec environment data.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Configuring AAA for Cisco TrustSec

You can use Cisco Secure ACS for Cisco TrustSec authentication. You must configure RADIUS server groups and specify the default AAA authentication and authorization methods on one of the Cisco TrustSec-enabled Cisco NX-OS devices in your network cloud. Because Cisco TrustSec supports RADIUS relay, you need to configure AAA only on a seed Cisco NX-OS device that is directly connected to a Cisco Secure ACS. For all the other Cisco TrustSec-enabled Cisco NX-OS devices, Cisco TrustSec automatically provides a private AAA server group, aaa-private-sg. The seed Cisco NX-OS devices uses the management virtual routing and forwarding (VRF) instance to communicate with the Cisco Secure ACS.

Note Only the Cisco Secure ACS supports Cisco TrustSec.

Configuring AAA on a Seed Cisco NX-OS Device in a Cisco TrustSec Network

This section describes how to configure AAA on the seed Cisco NX-OS device in your Cisco TrustSec network cloud.

Note When you configure the AAA RADIUS server group for the seed Cisco NX-OS device, you must specify a VRF instance. If you use the management VRF instance, no further configuration is necessary for the nonseed devices in the network cloud. If you use a different VRF instance, you must configure the nonseed devices with that VRF instance.

Before you begin

- Obtain the IPv4 or IPv6 address or hostname for the Cisco Secure ACS.
- Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** [**0** | **7**] *key* **pac** **Example:** `switch(config)# radius-server host 10.10.1.1 key L1a0K2s9 pac`	Configures a RADIUS server host with a key and PAC. The *hostname* argument is alphanumeric, case sensitive, and has a maximum of 256 characters. The *key* argument is alphanumeric, case sensitive, and has a maximum length of 63 characters. The **0** option indicates that the key is in clear text. The **7** option indicates that the key is encrypted. The default is clear text.
Step 3	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

Step 4	**aaa group server radius** *group-name*
	Example:
	`switch(config)# aaa group server radius Rad1`
	`switch(config-radius)#`
		Specifies the RADIUS server group and enters RADIUS server group configuration mode.

Step 5	**server** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** `switch(config-radius)# server 10.10.1.1`	Specifies the RADIUS server host address.
Step 6	**use-vrf** *vrf-name* **Example:**	Specifies the management VRF instance for the AAA server group.

	Command or Action	Purpose
	`switch(config-radius)# use-vrf management`	**Note** If you use the management VRF instance, no further configuration is necessary for the nonseed devices in the network cloud. If you use a different VRF instance, you must configure the nonseed devices with that VRF instance.

Step 7	**exit**
	Example:
	`switch(config-radius)# exit`
	`switch(config)#`
		Exits RADIUS server group configuration mode.

Step 8	**aaa authentication dot1x default group** *group-name* **Example:** `switch(config)# aaa authentication dot1x default group Rad1`	Specifies the RADIUS server groups to use for 802.1X authentication.
Step 9	**aaa authorization cts default group** *group-name* **Example:** `switch(config)# aaa authentication cts default group Rad1`	Specifies the RADIUS server groups to use for Cisco TrustSec authorization.

Step 10	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 11	(Optional) **show radius-server groups** [*group-name*] **Example:** `switch# show radius-server group rad1`	Displays the RADIUS server group configuration.
Step 12	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the AAA authentication configuration.
Step 13	(Optional) **show aaa authorization** **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration.
Step 14	(Optional) **show cts pacs** **Example:** `switch# show cts pacs`	Displays the Cisco TrustSec PAC information.

	Command or Action	Purpose
Step 15	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113
Configuring AAA on Cisco TrustSec Nonseed Cisco NX-OS Devices

Configuring Cisco TrustSec Authentication in Manual Mode

You can manually configure Cisco TrustSec on an interface if your Cisco NX-OS device does not have access to a Cisco Secure ACS or authentication is not needed because you have the MAC address authentication bypass feature enabled. You must manually configure the interfaces on both ends of the connection.

Note You cannot enable Cisco TrustSec on interfaces in half-duplex mode. Use the **show interface** command to determine if an interface is configured for half-duplex mode.

Caution For the Cisco TrustSec manual mode configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose

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

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

Step 3	**cts manual** **Example:**	Enters Cisco TrustSec manual configuration mode.

	Command or Action	Purpose
	`switch(config-if)# cts manual` `switch(config-if-cts-manual)#`	**Note** You cannot enable Cisco TrustSec on interfaces in half-duplex mode.
Step 4	(Optional) **policy dynamic identity** *peer-name* **Example:** `switch(config-if-cts-manual)# policy dynamic identity MyDevice2`	Configures a dynamic authorization policy download. The *peer-name* argument is the Cisco TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Cisco TrustSec credentials and AAA for Cisco TrustSec. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 5	(Optional) **policy static sgt** *tag* [**trusted**] **Example:** `switch(config-if-cts-manual)# policy static sgt 0x2`	Configures a static authorization policy. The *tag* argument is a hexadecimal value in the format **0x***hhhh*. The range is from 0x2 to 0xffef. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.

Step 6	**exit**
	Example:
	`switch(config-if-cts-manual)# exit`
	`switch(config-if)#`
		Exits Cisco TrustSec manual configuration mode.

Step 7	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 8	**no shutdown** **Example:**	Enables the interface and enables Cisco TrustSec authentication on the interface.

	Command or Action	Purpose
	`switch(config-if)# no shutdown`	

Step 9	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 10	(Optional) **show cts interface** {**all** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Cisco TrustSec configuration for the interfaces.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Configuring SGACL Policies

This section provides information about the configuration tasks for SGACL policies.

SGACL Policy Configuration Process

Follow these steps to configure Cisco TrustSec SGACL policies:

Procedure

Step 1 To improve performance, globally enable SGACL batch programming.

Step 2 For Layer 2 interfaces, enable SGACL policy enforcement for the VLANs with Cisco TrustSec-enabled interfaces.

Step 3 For Layer 3 interfaces, enable SGACL policy enforcement for the VRF instances with Cisco TrustSec-enabled interfaces.

Step 4 If you are not using AAA on a Cisco Secure ACS to download the SGACL policy configuration, manually configure the SGACL mapping and policies.

Enabling SGACL Policy Enforcement on VLANs

If you use SGACLs, you must enable SGACL policy enforcement in the VLANs that have Cisco TrustSec-enabled Layer 2 interfaces.

Note This operation cannot be performed on FCoE VLANs.

Before you begin

- Ensure that you enabled Cisco TrustSec.
- Ensure that you enabled SGACL batch programming.

Procedure

	Command or Action	Purpose

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

Step 2	**vlan** *vlan-id*
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Specifies a VLAN and enters VLAN configuration mode.

Step 3	**cts role-based enforcement** **Example:** `switch(config-vlan)# cts role-based enforcement`	Enables Cisco TrustSec SGACL policy enforcement on the VLAN. **Note** If you enable the cts role-based enforcement on a VLAN and no other configuration on ports, the traffic traversing through these ports are subject to (0,0) SGACL. You can either configure this SGACL statically or download it from Cisco ISE.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Saves the VLAN configuration and exits VLAN configuration mode.

Step 5	(Optional) **show cts role-based enable** **Example:** `switch(config)# show cts role-based enable`	Displays the Cisco TrustSec SGACL enforcement 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.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Manually Configuring Cisco TrustSec SGTs

You can manually configure unique Cisco TrustSec security group tags (SGTs) for the packets originating from this device.

Note You must also configure the Cisco TrustSec credentials for the Cisco NX-OS device on the Cisco Secure ACS.

Before you begin

Ensure that you have enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sgt** *tag* **Example:** `switch(config)# cts sgt 0x00a2`	Configures the SGT for packets sent from the device. The *tag* argument is a hexadecimal value in the format **0x***hhhh*. The range is from 0x2 to 0xffef.

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

Step 4	(Optional) **show cts environment-data** **Example:** `switch# show cts environment-data`	Displays the Cisco TrustSec environment data information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VLAN

You can manually configure an IPv4 address to SGACL SGT mapping on a VLAN if you do not have Cisco Secure ACS, dynamic ARP inspection, or DHCP snooping available on your Cisco NX-OS device.

Before you begin

- Ensure that you enabled Cisco TrustSec.
- Ensure that you enabled SGACL policy enforcement on the VLAN.

Procedure

	Command or Action	Purpose

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

Step 2	**vlan** *vlan-id*
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Specifies a VLAN and enters VLAN configuration mode.

Step 3	**cts role-based sgt-map** *ipv4-address tag* **Example:** `switch(config-vlan)# cts role-based sgt-map 10.10.1.1 100`	Configures SGT mapping for the SGACL policies for the VLAN.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Saves the VLAN configuration and exits VLAN configuration mode.

Step 5	(Optional) **show cts role-based sgt-map** **Example:** `switch(config)# show cts role-based sgt-map`	Displays the Cisco TrustSec SGACL SGT mapping 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.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113
Enabling SGACL Policy Enforcement on VLANs , on page 120
Enabling SGACL Policy Enforcement on VRF Instances

Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VRF Instance

You can manually configure IPv4-address-to-SGACL SGT mapping on a VRF instance if a Cisco Secure ACS is not available to download the SGACL policy configuration. You can use this feature if you do not have Cisco Secure ACS, dynamic ARP inspection, or DHCP snooping available on your Cisco NX-OS device.

Before you begin

- Ensure that you enabled Cisco TrustSec.
- Ensure that you enabled SGACL policy enforcement on the VRF instance.
- Ensure that the Layer-3 module is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context accounting`
	`switch(config-vrf)#`
		Specifies a VRF instance and enters VRF configuration mode.

Step 3	**cts role-based sgt-map** *ipv4-address tag* **Example:** `switch(config-vrf)# cts role-based sgt-map 10.10.1.1 100`	Configures SGT mapping for the SGACL policies for the VLAN.

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

Step 5	(Optional) **show cts role-based sgt-map** **Example:** `switch(config)# show cts role-based sgt-map`	Displays the Cisco TrustSec SGACL SGT mapping 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.

Manually Configuring SGACL Policies

You can manually configure SGACL policies on your Cisco NX-OS device if a Cisco Secure ACS is not available to download the SGACL policy configuration.

Before you begin

Ensure that you have enabled Cisco TrustSec.

For Cisco TrustSec logging to function, you must enable Cisco TrustSec counters or statistics.

Ensure that you have enabled SGACL policy enforcement on the VLAN and VRF instance.

Procedure

	Command or Action	Purpose

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

Step 2	**cts role-based access-list** *list-name*
	Example:
	`switch(config)# cts role-based access-list MySGACL`
	`switch(config-rbacl)#`
		Specifies an SGACL and enters role-based access list configuration mode. The *list-name* argument value is alphanumeric, case sensitive, and has a maximum length of 32 characters.

Step 3	(Optional) {**deny** | **permit**} **all** **Example:** `switch(config-rbacl)# deny all`	Denies or permits all traffic.
Step 4	(Optional) {**deny** | **permit**} **icmp** **Example:** `switch(config-rbacl)# permit icmp`	Denies or permits Internet Control Message Protocol (ICMP) traffic.
Step 5	(Optional) {**deny** | **permit**} **igmp** **Example:** `switch(config-rbacl)# deny igmp`	Denies or permits Internet Group Management Protocol (IGMP) traffic.
Step 6	(Optional) {**deny** | **permit**} **ip** **Example:** `switch(config-rbacl)# permit ip`	Denies or permits IP traffic.
Step 7	(Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# deny tcp dst eq 100`	Denies or permits TCP traffic. The default permits all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.

	Command or Action	Purpose
Step 8	{**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# permit udp src eq 1312`	Denies or permits UDP traffic. The default permits all UDP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.

Step 9	**exit**
	Example:
	`switch(config-rbacl)# exit`
	`switch(config)#`
		Exits role-based access-list configuration mode.

Step 10	**cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name* **Example:** `switch(config)# cts role-based sgt 3 dgt 10 access-list MySGACL`	Maps the SGT values to the SGACL. The *sgt-value* and *dgt-value* argument values range from 0 to 65520. **Note** You must create the SGACL before you can map SGTs to it.
Step 11	(Optional) **show cts role-based access-list** **Example:** `switch(config)# show cts role-based access-list`	Displays the Cisco TrustSec SGACL configuration.
Step 12	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113
Enabling SGACL Policy Enforcement on VLANs , on page 120
Enabling SGACL Policy Enforcement on VRF Instances

Displaying the Downloaded SGACL Policies

After you configure the Cisco TrustSec device credentials and AAA, you can verify the Cisco TrustSec SGACL policies downloaded from the Cisco Secure ACS. The Cisco NX-OS software downloads the SGACL policies when it learns of a new SGT through authentication and authorization on an interface, from SXP, or from manual IPv4 address to SGACL SGT mapping.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	**show cts role-based access-list** **Example:** `switch# show cts role-based access-list`	Displays Cisco TrustSec SGACLs, both downloaded from the Cisco Secure ACS and manually configured on the Cisco NX-OS device.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Refreshing the Downloaded SGACL Policies

You can refresh the SGACL policies downloaded to the Cisco NX-OS device by the Cisco Secure ACS.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	**cts refresh role-based-policy** **Example:** `switch# cts refresh role-based-policy`	Refreshes the Cisco TrustSec SGACL policies from the Cisco Secure ACS.
Step 2	(Optional) **show cts role-based policy** **Example:** `switch# show cts role-based policy`	Displays the Cisco TrustSec SGACL policies.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Enabling CTS Batched Programming

In order to program a large number of SGT, DGT pairs (usually greater than 100) manually or by using ISE when RBACL enforcement is enabled on VLANs, batched programming must be enabled for faster programming and improved performance.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts role-based batched-programming**	Enables CTS batched programming for faster programming of SGACLs associated with large numbers of SGT, DGT pairs.

	Command or Action	Purpose
		Note Use the **no** form of this command to disable cts role-based batched programming. We do not recommend disabling the **cts role-based batched-programming** command if you have greater than 100 SGT, DGT pairs with RBACL enforcement enabled on VLANs.

Example

This example shows how to enable CTS batched programming:

```
switch# configure terminal
switch(config)# cts role-based batched-programming
```

Enabling Statistics for RBACL

You can request a count of the number of packets that match role-based access control list (RBACL) policies. These statistics are collected per source group tag (SGT) and destination group tag (DGT).

Note When you modify an RBACL policy, statistics for the previously assigned access control entry (ACE) are displayed, and the newly assigned ACE statistics are initialized to 0.

Note RBACL statistics are lost only when the Cisco NX-OS device reloads or you deliberately clear the statistics.

Before you begin

Ensure that you have enabled Cisco TrustSec.

If you plan to enable RBACL statistics, ensure that you have enabled RBACL policy enforcement on the VLAN and VRF instance.

When you enable RBACL statistics, each policy requires one entry in the hardware. If you do not have enough space remaining in the hardware, an error message appears, and you are unable to enable the statistics.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	[**no**] **cts role-based counters enable** **Example:** `switch(config)# cts role-based counters enable`	Enables or disables RBACL statistics. The default is disabled.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 5	(Optional) **show cts role-based counters** [**sgt** {*sgt-value* | **any** | **unknown**}] [**dgt** {*dgt-value* | **any** | **unknown**}] **Example:** `switch# show cts role-based counters sgt 10 dgt 20`	Displays the configuration status of RBACL statistics and lists statistics for all RBACL policies. Optionally displays the total number of packets that match RBACL policies for a specific source group tag (SGT) or destination group tag (DGT). The *sgt-value* and *dgt-value* argument values range from 0 to 65519.
Step 6	(Optional) **clear cts role-based counters** **Example:** `switch# clear cts role-based counters`	Clears the RBACL statistics so that all counters are reset to 0.

Clearing Cisco TrustSec SGACL Policies

You can clear the Cisco TrustSec SGACL policies.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show cts role-based policy** **Example:** `switch# clear cts policy all`	Displays the Cisco TrustSec RBACL policy configuration.
Step 2	**clear cts policy** {**all** | **peer** *device-name* | **sgt** *sgt-value*} **Example:**	Clears the policies for Cisco TrustSec connection information.

	Command or Action	Purpose
	`switch# clear cts policy all`	

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Manually Configuring SXP

You can use the SGT Exchange Protocol (SXP) to propagate the SGTs across network devices that do not have hardware support for Cisco TrustSec. This section describes how to configure Cisco TrustSec SXP on Cisco NX-OS devices in your network.

Cisco TrustSec SXP Configuration Process

Follow these steps to manually configure Cisco TrustSec SXP:

Procedure

Step 1 Enable the Cisco TrustSec feature.

Step 2 Enable SGACL policy enforcement on the VRF instance.

Step 3 Enable Cisco TrustSec SXP.

Step 4 Configure SXP peer connections.

Note You cannot use the management (mgmt 0) connection for SXP.

Related Topics

Enabling SGACL Policy Enforcement on VLANs , on page 120
Enabling SGACL Policy Enforcement on VRF Instances
Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VLAN, on page 123
Manually Configuring SGACL Policies, on page 125
Enabling the Cisco TrustSec SGT Feature , on page 113
Enabling Cisco TrustSec SXP , on page 130
Configuring Cisco TrustSec SXP Peer Connections, on page 131

Enabling Cisco TrustSec SXP

You must enable Cisco TrustSec SXP before you can configure peer connections.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sxp enable** **Example:** `switch(config)# cts sxp enable`	Enables SXP for Cisco TrustSec.

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

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Configuring Cisco TrustSec SXP Peer Connections

You must configure the SXP peer connection on both the speaker and listener devices. When using password protection, make sure to use the same password on both ends.

Note If the default SXP source IP address is not configured and you do not specify the SXP source address in the connection, the Cisco NX-OS software derives the SXP source IP address from existing local IP addresses. The SXP source address could be different for each TCP connection initiated from the Cisco NX-OS device.

Before you begin

Ensure that you enabled Cisco TrustSec.

Ensure that you enabled SXP.

Ensure that you enabled RBACL policy enforcement in the VRF instance.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sxp connection peer** *peer-ipv4-addr* [**source** *src-ipv4-addr*] **password** {**default** | **none** | **required** *password*} **mode** {**speaker** | **listener** } [**vrf** *vrf-name*] **Example:** `switch(config)# cts sxp connection peer 10.10.1.1 source 20.20.1.1 password default mode listener`	Configures the SXP address connection. The **source** keyword specifies the IPv4 address of the source device. The default source is IPv4 address you configured using the **cts sxp default source-ip** command. The **password** keyword specifies the password that SXP should use for the connection using the following options: • Use the **default** option to use the default SXP password that you configured using the **cts sxp default password** command. • Use the **none** option to not use a password. • Use the **required** option to use the password specified in the command. The **speaker** and **listener** keywords specify the role of the remote peer device. The **vrf** keyword specifies the VRF instance to the peer. The default is the default VRF instance. **Note** You cannot use the management (mgmt 0) interface for SXP.

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

Step 4	(Optional) **show cts sxp connections** **Example:** `switch# show cts sxp connections`	Displays the SXP connections and their status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113
Enabling Cisco TrustSec SXP , on page 130
Enabling SGACL Policy Enforcement on VRF Instances

Configuring the Default SXP Password

By default, SXP uses no password when setting up connections. You can configure a default SXP password for the Cisco NX-OS device.

Before you begin

Ensure that you enabled Cisco TrustSec.

Ensure that you enabled SXP.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sxp default password** *password* **Example:** `switch(config)# cts sxp default password A2Q3d4F5`	Configures the SXP default password.

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

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **show running-config cts** **Example:** `switch# show running-config cts`	Displays the SXP configuration in the running configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Enabling Cisco TrustSec SXP , on page 130

Configuring the Default SXP Source IPv4 Address

The Cisco NX-OS software uses the default source IPv4 address in all new TCP connections where a source IPv4 address is not specified. When you change the default source IP address, the existing SXP connections are reset and the IP-SGT bindings learned over SXP are cleared. The SXP connections, for which a source IP address has been configured, will continue to use the same IP address, while coming back up.

The SXP connections, for which a source IP address has not been configured, uses the default IP address as the source IP address. Note that for such connections, correct destination IP address configuration on the peer and the reachability to the default source IP address are the required conditions before such connections can become operational. It is recommended to ensure that these conditions are met for existing operational connections, before configuring default source IP address on a device.

Before you begin

Ensure that you enabled Cisco TrustSec.

Ensure that you enabled SXP.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sxp default source-ip** *src-ip-addr* **Example:** `switch(config)# cts sxp default source-ip 10.10.3.3`	Configures the SXP default source IPv4 address.

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

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113

Enabling Cisco TrustSec SXP , on page 130

Changing the SXP Retry Period

The SXP retry period determines how often the Cisco NX-OS software retries an SXP connection. When an SXP connection is not successfully set up, the Cisco NX-OS software makes a new attempt to set up the connection after the SXP retry period timer expires. The default value is 60 seconds (1 minute). Setting the SXP retry period to 0 seconds disables the timer and retries are not attempted.

Before you begin

Ensure that you enabled Cisco TrustSec.

Ensure that you enabled SXP.

Procedure

	Command or Action	Purpose

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

Step 2	**cts sxp retry-period** *seconds* **Example:** `switch(config)# cts sxp retry-period 120`	Changes the SXP retry timer period. The default value is 60 seconds (1 minute). The range is from 0 to 64000.

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

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Cisco TrustSec SGT Feature , on page 113
Enabling Cisco TrustSec SXP , on page 130

Verifying the Cisco TrustSec Configuration

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

Command	Purpose
show cts	Displays Cisco TrustSec information.
show cts credentials	Displays Cisco TrustSec credentials for EAP-FAST.
show cts environment-data	Displays Cisco TrustSec environmental data.
show cts interface {**all** | **ethernet** *slot*/*port*}	Displays the Cisco TrustSec configuration for the interfaces.
show cts role-based access-list	Displays Cisco TrustSec SGACL information.
show cts pacs	Displays Cisco TrustSec authorization information and PACs in the device key store.
show cts role-based enable	Displays Cisco TrustSec SGACL enforcement status.
show cts role-based policy	Displays Cisco TrustSec SGACL policy information.
show cts role-based sgt-map	Displays the Cisco TrustSec SGACL SGT map configuration.
show cts sxp	Displays Cisco TrustSec SXP information.
show running-config cts	Displays the Cisco TrustSec information in the running configuration.

Configuration Examples for Cisco TrustSec

This section provides configuration examples for Cisco TrustSec.

Example: Enabling Cisco TrustSec

The following example shows how to enable Cisco TrustSec:

```
feature dot1x
feature cts
cts device-id device1 password Cisco321
```

Example: Configuring AAA for Cisco TrustSec on a Seed Cisco NX-OS Device

The following example shows how to configure AAA for Cisco TrustSec on the seed Cisco NX-OS device:

```
radius-server host 10.10.1.1 key Cisco123 pac
aaa group server radius Rad1
  server 10.10.1.1
```

```
  use-vrf management
aaa authentication dot1x default group Rad1
aaa authorization cts default group Rad1
```

Example: Configuring Cisco TrustSec Authentication in Manual Mode

The following example shows how to configure Cisco TrustSec authentication in manual mode static policy on an interface:

```
interface ethernet 2/1
  cts manual

    policy static sgt 0x20
```

The following example shows how to configure Cisco TrustSec authentication in manual mode dynamic policy on an interface:

```
interface ethernet 2/2
  cts manual
    policy dynamic identity device2
```

Example: Configuring Cisco TrustSec Role-Based Policy Enforcement for a VLAN

The following example shows how to enable Cisco TrustSec role-based policy enforcement for a VLAN:

```
vlan 10
  cts role-based enforcement
```

Example: Configuring IPv4 Address to SGACL SGT Mapping for the Default VRF Instance

The following example shows how to manually configure IPv4 address to SGACL SGT mapping for Cisco TrustSec role-based policies for the default VRF instance:

```
cts role-based sgt-map 10.1.1.1 20
```

Example: Configuring IPv4 Address to SGACL SGT Mapping for a VLAN

The following example shows how to manually configure IPv4 address to SGACL SGT mapping for Cisco TrustSec role-based policies for a VLAN:

```
vlan 10
  cts role-based sgt-map 20.1.1.1 20
```

Example: Manually Configuring Cisco TrustSec SGACLs

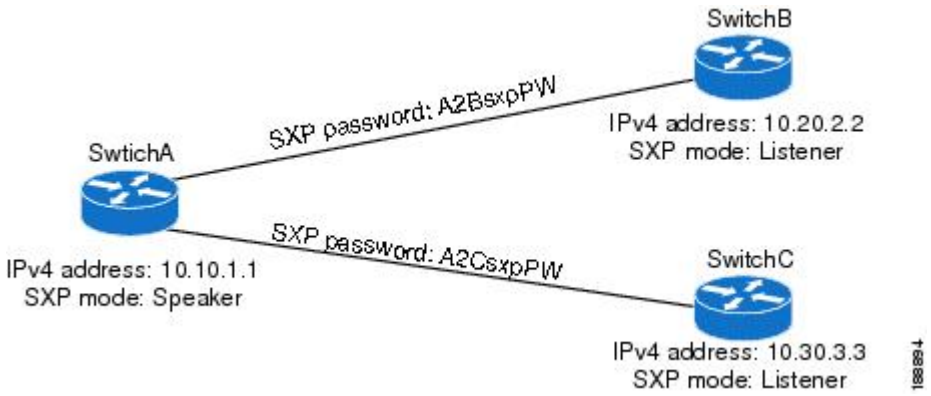
The following example shows how to manually configure Cisco TrustSec SGACLs:

```
cts role-based access-list abcd
  permit icmp
cts role-based sgt 10 dgt 20 access-list abcd
```

Example: Manually Configuring SXP Peer Connections

This figure shows an example of SXP peer connections over the default VRF instance.

Figure 11: Example SXP Peer Connections



The following example shows how to configure the SXP peer connections on SwitchA:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.20.2.2 password required A2BsxpPW mode listener
cts sxp connection peer 10.30.3.3 password required A2CsxpPW mode listener
```

The following example shows how to configure the SXP peer connection on SwitchB:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.10.1.1 password required A2BsxpPW mode speaker
```

The following example shows how to configure the SXP peer connection on SwitchC:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.10.1.1 password required A2CsxpPW mode speaker
```

Additional References for Cisco TrustSec

This sections provides additional information related to implementing Cisco TrustSec.

Related Documentation

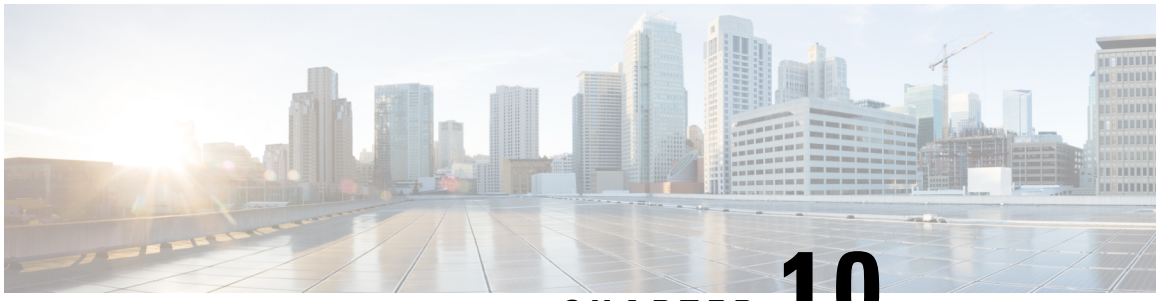
Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
Command Reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference*

Feature History for Cisco TrustSec

This table lists the release history for this feature.

Table 13: Feature History for Cisco TrustSec

Feature Name	Releases	Feature Information	
Cisco TrustSec	6.1(1)	Removed the requirement for the Advanced Services license.	
Cisco TrustSec	6.1(1)	Added MACsec support for 40G and 100G M2 Series modules.	
Cisco TrustSec	5.2(1)	Supports pause frame encryption and decryption on interfaces.	
SGACL policies	5.0(2)	Supports the enabling or disabling of RBACL logging.	
SGACL policies	5.0(2)	Supports the enabling, disabling, monitoring, and clearing of RBACL statistics.	
Cisco TrustSec	4.2(1)	No change from Release 4.1.	



CHAPTER 10

Configuring Access Control Lists

This chapter contains the following sections:

Information About ACLs

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

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

IP ACL Types and Applications

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

Table 14: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	IPv4 ACLs IPv6 ACLs
Router ACL	• VLAN interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	IPv4 ACLs IPv6 ACLs
VLAN ACL (VACL)	An ACL is a VACL when you use an access map to associate the ACL with an action and then apply the map to a VLAN.	IPv4 ACLs
VTY ACL	VTYs	IPv4 ACLs IPv6 ACLs

Application Order

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

1. Port ACL
2. Ingress VACL
3. Ingress Router ACL
4. Egress Router ACL
5. Egress VACL

Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

he device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a 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


permit icmp any any nd-na
permit icmp any any nd-ns
permit icmp any any router-advertisement
permit icmp any any router-solicitation
```

Unless you configure an IPv6 ACL with a rule that denies ICMPv6 neighbor discovery messages, the first four rules ensure that the device permits neighbor discovery advertisement and solicitation messages. The fifth rule ensures that the device denies unmatched IPv6 traffic.

Note If you explicitly configure an IPv6 ACL with a **deny ipv6 any any** rule, the implicit permit rules can never permit traffic. If you explicitly configure a **deny ipv6 any any** rule but want to permit ICMPv6 neighbor discovery messages, explicitly configure a rule for all five implicit rules.

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

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

Additional Filtering Options

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

- Layer 4 protocol
- TCP and UDP ports
- ICMP types and codes
- IGMP types
- Precedence level
- Differentiated Services Code Point (DSCP) value
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections

MAC ACLs support the following additional filtering options:

- Layer 3 protocol
- VLAN ID
- Class of Service (CoS)

Sequence Numbers

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

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

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

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

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

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

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

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

Logical Operators and Logical Operation Units

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

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

Note The range operator is inclusive of boundary values.

These LOUs minimize the number of ternary content addressable memory (TCAM) entries needed to perform these operations. A maximum of two LOUs are allowed for each feature on an interface. For example an ingress RACL can use two LOUs, and a QoS feature can use two LOUs. If an ACL feature requires more than two arithmetic operations, the first two operations use LOUs, and the remaining access control entries (ACEs) get expanded.

The following guidelines determine when the device stores operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples "gt 10" and "gt 11" would be stored separately in half an LOU each. The couples "gt 10" and "lt 10" would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

Policy-Based ACLs

The device supports policy-based ACLs (PBACLs), which allow you to apply access control policies across object groups. An object group is a group of IP addresses or a group of TCP or UDP ports. When you create a rule, you specify the object groups rather than specifying IP addresses or ports.

Using object groups when you configure IPv4 or IPv6 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses or ports from the source or destination of rules. For example, if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the device expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the I/O module when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to port, router, and VLAN ACLs:

IPv4 address object groups

Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

IPv6 address object groups

Can be used with IPv6 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

Protocol port object groups

Can be used with IPv4 and IPv6 TCP and UDP rules to specify source or destination ports. When you use the **permit** or **deny** command to configure a rule, the **portgroup** keyword allows you to specify an object group for the source or destination.

Statistics and ACLs

The device can maintain global statistics for each rule that you configure in IPv4, IPv6, and MAC ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

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

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

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

Licensing Requirements for ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	No license is required to use ACLs.

Prerequisites for ACLs

IP ACLs have the following prerequisites:

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

VACLs have the following prerequisite:

- Ensure that the IP ACL that you want to use in the VACL exists and is configured to filter traffic in the manner that you need for this application.

Guidelines and Limitations for ACLs

IP ACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally.

MAC ACLs have the following configuration guidelines and limitations:

- MAC ACLs apply to ingress traffic only.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

VACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configurations using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

Default ACL Settings

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

Table 15: Default IP ACLs Parameters

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

The following table lists the default settings for VACL parameters.

Table 16: Default VACL Parameters

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

Configuring IP ACLs

Creating an IP ACL

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

Procedure

Step 1 switch# **configure terminal**

Enters global configuration mode.

Step 2 switch(config)# {**ip** | **ipv6**} **access-list** *name*

Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3 switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*

Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295.

The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for the specific Cisco Nexus device.

Step 4 (Optional) switch(config-acl)# **statistics**

Specifies that the switch maintains global statistics for packets that match the rules in the ACL.

Step 5 (Optional) switch# **show** {**ip** | **ipv6**} **access-lists** *name*

Displays the IP ACL configuration.

Step 6 (Optional) switch# **copy running-config startup-config**

Copies the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 ACL:

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

```
switch(config-acl)# statistics
```

This example shows how to create an IPv6 ACL:

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

Changing an IP ACL

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

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

Procedure

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

	Command or Action	Purpose
Step 7	(Optional) switch#**show ip access-lists** *name*	Displays the IP ACL configuration.
Step 8	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 150

Removing an IP ACL

You can remove an IP ACL from the switch.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no** {**ip** | **ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration. The removed IP ACL should not appear.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(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.

Configuring ACLs with Logging

You can create an access-control list for logging traffic of a specified protocol and address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip** | **ipv6**} **access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# **permit** *protocol source destination* **log**	Creates a rule to log traffic of the specified protocol in the syslog file. in the IP ACL. Valid values for the *protocol* argument are: • **icmp**—ICMP • **igmp**—IGMP • **ip**—IPv4 • **ipv6**—IPv6 • **tcp**—TCP • **udp**—UDP • **sctp**—SCTP (IPv6 only) The source and destination arguments can be the IP address with a network wildcard (IPv4 only), IP address and variable-length subnet mask, host address, or **any** to designate any address. For more information, see the System Management configuration guide and the Security command reference for your platform.
Step 4	switch(config-acl)# **exit**	Exists the current configuration mode.
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 shows how to create an ACL for logging entries that match IPv4 TCP traffic from any source and any destination:

```
switch# configuration terminal
switch(config)# ip access-list tcp_log
switch(config-acl)# permit tcp any any log
```

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

Applying an IP ACL to mgmt0

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

Step 2	**interface mgmt** *port*
	Example:
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Enters configuration mode for the management interface.

Step 3	**ip access-group** *access-list* {**in** | **out**} **Example:** `switch(config-if)#ip access-group acl-120 out`	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

- Creating an IP ACL

Applying an IP ACL as a Router ACL

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

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

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 tunnel** *tunnel-number* • switch(config)# **interface vlan** *vlan-ID* • switch(config)# **interface mgmt** *port*	Enters configuration mode for the interface type that you specified.
Step 3	Enter one of the following commands: • switch(config-if)# **ip access-group** *access-list* {**in** | **out**} • switch(config-if)# **ipv6 traffic-filter** *access-list* {**in** | **out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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.

Verifying IP ACL Configurations

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

Procedure

- switch# **show running-config**

  Displays ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.

- switch# **show running-config interface**

  Displays the configuration of an interface to which you have applied an ACL.

Example

For detailed information about the fields in the output from these commands, refer to the *Command Reference* for your Cisco Nexus device.

Monitoring and Clearing IP ACL Statistics

Use the **show ip access-lists** or **show ipv6 access-list** command to display statistics about an IP ACL, including the number of packets that have matched each rule. For detailed information about the fields in the output from this command, see the *Command Reference* for your Cisco Nexus device.

Note The mac access-list is applicable to non-IPv4 and non-IPv6 traffic only.

Procedure

- switch# **show** {**ip** | **ipv6**} **access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** and **show ipv6 access-list** command output includes the number of packets that have matched each rule.

- switch#**show ip access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.

- switch# **clear** {**ip** | **ipv6**} **access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

- switch# **clear ip access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

Configuring Object Groups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL and IPv6 ACL rules.

Session Manager Support for Object Groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

Procedure

	Command or Action	Purpose

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

Step 2	**object-group ip address** *name*
	Example:
	`switch(config)# object-group ip address ipv4-addr-group-13`
	`switch(config-ipaddr-ogroup)#`
		Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.

	Command or Action	Purpose
Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv4-address* • [*sequence-number*] *IPv4-address network-wildcard* • [*sequence-number*] *IPv4-address*/*prefix-len* **Example:** `switch(config-ipaddr-ogroup)# host 10.99.32.6`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command to specify a network of hosts.
Step 4	Enter one of the following commands: • **no** [*sequence-number*] • **no host** *IPv4-address* • **no** *IPv4-address network-wildcard* • **no** *IPv4-address*/*prefix-len* **Example:** `switch(config-ipaddr-ogroup)# no host 10.99.32.6`	Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipaddr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating and Changing an IPv6 Address Object Group

You can create and change an IPv6 address group object.

Procedure

	Command or Action	Purpose

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

Step 2	**object-group ipv6 address** *name* **Example:**	Creates the IPv6 address object group and enters IPv6 address object-group configuration mode.

	Command or Action	Purpose
	`switch(config)# object-group ipv6 address ipv6-addr-group-A7` `switch(config-ipv6addr-ogroup)#`	
Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv6-address* • [*sequence-number*] *IPv6-address*/*prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command specify a network of hosts.
Step 4	Enter one of the following commands: • **no** *sequence-number* • **no host** *IPv6-address* • **no** *IPv6-address*/*prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# no host 2001:db8:0:3ab0::1`	Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipv6addr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating and Changing a Protocol Port Object Group

You can create and change a protocol port object group.

Procedure

	Command or Action	Purpose

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

Step 2	**object-group ip port** *name* **Example:**	Creates the protocol port object group and enters port object-group configuration mode.

	Command or Action	Purpose
	`switch(config)# object-group ip port NYC-datacenter-ports` `switch(config-port-ogroup)#`	
Step 3	[*sequence-number*] *operator port-number* [*port-number*] **Example:** `switch(config-port-ogroup)# eq 80`	Creates an entry in the object group. For each entry that you want to create, use one of the following operator commands: • **eq**—Matches the port number that you specify only. • **gt**—Matches port numbers that are greater than (and not equal to) the port number that you specify. • **lt**—Matches port numbers that are less than (and not equal to) the port number that you specify. • **neq**—Matches all port numbers except for the port number that you specify. • **range**—Matches the range of port number between and including the two port numbers that you specify. **Note** The **range** command is the only operator command that requires two *port-number* arguments.
Step 4	**no** {*sequence-number* | *operator port-number* [*port-number*]} **Example:** `switch(config-port-ogroup)# no eq 80`	Removes an entry from the object group. For each entry that you want to remove, use the **no** form of the applicable operator command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing an Object Group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

Procedure

	Command or Action	Purpose

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

Step 2	**no object-group** {**ip address** | **ipv6 address** | **ip port**} *name* **Example:** `switch(config)# no object-group ip address ipv4-addr-group-A7`	Removes the object group that you specified.
Step 3	(Optional) **show object-group** **Example:** `switch(config)# show object-group`	Displays all object groups. The removed object group should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Object-Group Configuration

To display object-group configuration information, perform one of the following tasks:

Command	Purpose
show object-group	Displays the object-group configuration.
show running-config aclmgr	Displays ACL configuration, including object groups.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference*.

Information About VLAN ACLs

A VLAN ACL (VACL) is one application of an IP ACL. You can configure VACLs to apply to all packets that are bridged within a VLAN. VACLs are used strictly for security packet filtering. VACLs are not defined by direction (ingress or egress).

VACLs and Access Maps

VACLs use access maps to link an IP ACL to an action. The switch takes the configured action on packets that are permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.
- Drop—Drops the traffic.

Statistics

The Cisco Nexus device can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

Note The Cisco Nexus device does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the switch maintains statistics for that VACL. This allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

Configuring VACLs

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP ACL with an action to be applied to the matching traffic.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan access-map** *map-name*	Enters access map configuration mode for the access map specified.
Step 3	switch(config-access-map)# **match ip address** *ip-access-list*	Specifies an IPv4 and IPv6 ACL for the map.
Step 4	switch(config-access-map)# **action** {**drop** | **forward**}	Specifies the action that the switch applies to traffic that matches the ACL.
Step 5	(Optional) switch(config-access-map)# [**no**] **statistics**	Specifies that the switch maintains global statistics for packets matching the rules in the VACL. The **no** option stops the switch from maintaining global statistics for the VACL.

	Command or Action	Purpose
Step 6	(Optional) switch(config-access-map)# **show running-config**	Displays the ACL configuration.
Step 7	(Optional) switch(config-access-map)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

Be sure that you know whether the VACL is applied to a VLAN. The switch allows you to remove VACLs that are current applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the switch considers the removed VACL to be empty.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no vlan access-map** *map-name*	Removes the VLAN access map configuration for the specified access map.
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **vlan filter** *map-name* **vlan-list** *list*	Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. The **vlan-list** command can specify a list of up to 32 VLANs, but multiple **vlan-list** commands can be configured to cover more than 32 VLANs.
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.

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

Verifying VACL Configuration

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

Procedure

- switch# **show running-config aclmgr**

  Displays ACL configuration, including VACL-related configuration.

- switch# **show vlan filter**

  Displays information about VACLs that are applied to a VLAN.

- switch# **show vlan access-map**

  Displays information about VLAN access maps.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, perform one of the following tasks:

Procedure

- switch# **show vlan access-list**

  Displays VACL configuration. If the VLAN access-map includes the **statistics** command, then the **show vlan access-list** command output includes the number of packets that have matched each rule.

- switch# **clear vlan access-list counters**

  Clears statistics for all VACLs or for a specific VACL.

Configuration Examples for VACL

The following example shows how to configure a VACL to forward traffic permitted by an IP ACL named acl-ip-01 and how to apply the VACL to VLANs 50 through 82:

```
switch# configure terminal
switch(config)# vlan access-map acl-ip-map
switch(config-access-map)# match ip address acl-ip-01
switch(config-access-map)# action forward
switch(config-access-map)# exit
 switch(config)# vlan filter acl-ip-map vlan-list 50-82
```

Configuring ACLs on Virtual Terminal Lines

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

Follow these guidelines when configuring ACLs on VTY lines:

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

Before you begin

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

Procedure

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

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

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

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

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

	Command or Action	Purpose
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:

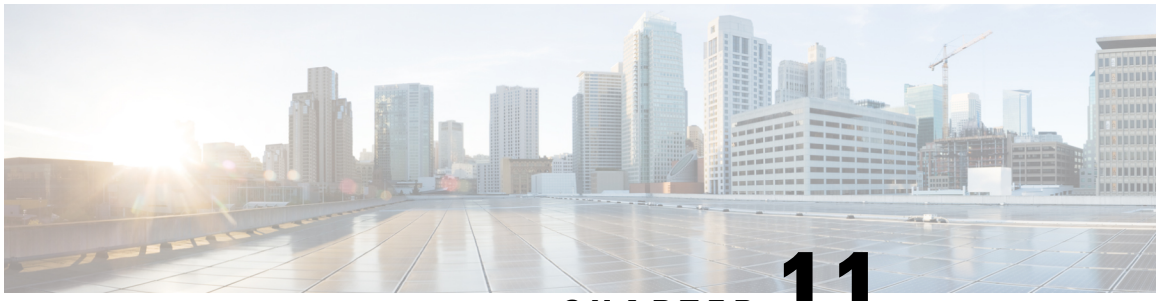
```
switch# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
switch(config)# ip access-list ozi2
switch(config-acl)# statistics per-entry
switch(config-acl)# deny tcp 172.18.217.83/32 any
switch(config-acl)# exit

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 11

Configuring Port Security

This chapter includes the following sections:

Information About Port Security

Port security allows you to configure Layer 2 physical interfaces and Layer 2 port-channel interfaces to allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

Note Unless otherwise specified, the term *interface* refers to both physical interfaces and port-channel interfacesphysical interfaces, port-channel interfaces, and vPCs; likewise, the term *Layer 2 interface* refers to both Layer 2 physical interfaces and Layer 2 port-channel interfaces.

Secure MAC Address Learning

The process of securing a MAC address is called learning. A MAC address can be a secure MAC address on one interface only. For each interface that you enable port security on, the device can learn a limited number

of MAC addresses by the static, dynamic, or sticky methods. The way that the device stores secure MAC addresses varies depending upon how the device learned the secure MAC address.

Static Method

The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are unaffected if the device restarts.

A static secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address from the configuration.
- You configure the interface to act as a Layer 3 interface.

Adding secure addresses by the static method is not affected by whether dynamic or sticky address learning is enabled.

Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains secured on an interface until one of the following events occurs:

- The device restarts.
- The interface restarts.
- The address reaches the age limit that you configured for the interface.
- You explicitly remove the address. For more information, see Removing a Dynamic Secure MAC Address, on page 181.
- You configure the interface to act as a Layer 3 interface.

Sticky Method

If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning, but the device stores addresses learned by this method in nonvolatile RAM (NVRAM). As a result, addresses learned by the sticky method persist through a device restart. Sticky secure MAC addresses do not appear in the running configuration of an interface.

Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, the device stops dynamic learning and performs sticky learning instead. If you disable sticky learning, the device resumes dynamic learning.

A sticky secure MAC address entry remains secured on an interface until one of the following events occurs:

- You explicitly remove the sticky MAC address configuration from the interface. For more information, see Removing a Sticky Secure MAC Address, on page 180.
- You configure the interface to act as a Layer 3 interface.

Dynamic Address Aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 1 to 1440 minutes. The default aging time is 0, which disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The two methods of determining address age are as follows:

Inactivity

The length of time after the device last received a packet from the address on the applicable interface.

Absolute

The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Note If the absolute method is used to age out a MAC address, then depending on the traffic rate, few packets may drop each time a MAC address is aged out and relearned. To avoid this use inactivity timeout.

Secure MAC Address Maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: dynamic, sticky, or static.

Note In vPC domains, the configuration on the primary vPC takes effect.

Tip To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

The following three limits can determine how many secure MAC addresses are permitted on an interface:

System maximum

The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

Interface maximum

You can configure a maximum number of 1025 secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Sum of all interface maximums on a switch cannot exceed the system maximum.

VLAN maximum

You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. The sum of all VLAN maximums under an interface cannot exceed the configured

interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first. Otherwise, the configuration of new limit is rejected.

Security Violations and Actions

Port security triggers security violations when either of the two following events occur:

MAX Count Violation

Ingress traffic arrives at an interface from a nonsecure MAC address and learning the address would exceed the applicable maximum number of secure MAC addresses.

When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

- VLAN 1 has a maximum of 5 addresses
- The interface has a maximum of 20 addresses

The device detects a violation when any of the following occurs:

- The device has learned five addresses for VLAN 1 and inbound traffic from a sixth address arrives at the interface in VLAN 1.
-

MAC Move Violation

Ingress traffic from a secure MAC address arrives at a different secured interface in the same VLAN as the interface on which the address is secured.

When a security violation occurs, the device increments the security violation counter for the interface and takes the action specified by the port security configuration of the interface. If a violation occurs because ingress traffic from a secure MAC address arrives at a different interface than the interface on which the address is secure, the device applies the action on the interface that received the traffic.

The violation modes and the possible actions that a device can take are as follows:

Shutdown violation mode

Error disables the interface that received the packet triggering the violation and the port shuts down. The security violation count is set to 1. This action is the default. After you reenable the interface, it retains its port security configuration, including its static and sticky secure MAC addresses. However, the dynamic MAC addresses are not retained and have to relearned.

You can use the **errdisable recovery cause pscecure-violation** global configuration command to configure the device to reenable the interface automatically if a shutdown occurs, or you can manually reenable the interface by entering the **shutdown** and **no shut down** interface configuration commands. For detailed information about the commands, see the Security Command Reference for your platform.

Restrict violation mode

Drops ingress traffic from any nonsecure MAC addresses.

The device keeps a count of the number of unique source MAC addresses of dropped packets, which is called the security violation count.

Violation is triggered for each unique nonsecure source MAC address and security violation count increments till 10, which is the maximum value. The maximum value of 10 is fixed and not configurable.

Address learning continues until the maximum security violations (10 counts) have occurred on the interface. Traffic from addresses learned after the first security violation are added as BLOCKED entries in the MAC table and dropped. These BLOCKED MAC address age out after 5 minutes. The BLOCKED MAC address age out time of 5 minutes is fixed and not configurable.

Depending on the violation type, RESTRICT mode action varies as follows:

- In case of MAX count violation, after the maximum number of MAX count violations (10) is reached, the device stops learning new MAC addresses. Interface remains up.
- In case of MAC move violation, when the maximum security violations have occurred on the interface, the interface is error Disabled.

Protect violation mode

Prevents further violations from occurring. The address that triggered the security violation is learned but any traffic from the address is dropped. Security violation counter is set to 1, which is the maximum value. Further address learning stops. Interface remains up.

Note that the security violation is reset to 0 after the interface is recovered from violation through one of the following events:

- Dynamic secure MAC addresses age out
- Interface flap, link down, or link up events
- Port-security disable and re-enable on the interface
- Changing violation mode of the interface

Note If an interface is errDisabled, you can bring it up only by flapping the interface.

Port Security and Port Types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

Access ports

You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN. VLAN maximums are not useful for access ports.

Trunk ports

You can configure port security on interfaces that you have configured as Layer 2 trunk ports. The device allows VLAN maximums only for VLANs associated with the trunk port.

SPAN ports

You can configure port security on SPAN source ports but not on SPAN destination ports.

Ethernet port channels

You can configure port security on Layer 2 Ethernet port channels in either access mode or trunk mode.

Virtual port channels

Port security is supported on orphan ports, switch virtual port channels (VPCs), straight-through vPCs, active-active VPCs, and enhanced Layer 2 vPCs.

Fabric Extender (FEX) ports

Port security is supported on GEM and FEX ports.

Private VLAN Enabled Ports

Port Security is supported on ports that are enabled as Private VLAN ports.

PVLAN Host (physical interfaces only)

You can configure Private VLANs (PVLANs) to provide traffic separation and security at the Layer 2 level. A PVLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN.

PVLAN Promiscuous (physical interfaces only)

You can configure a Layer 2 VLAN network interface, or switched virtual interface (SVI), on the PVLAN promiscuous port, which provides routing functionality to the primary PVLAN. This is supported on physical interfaces only.

PVLAN trunk secondary/promiscuous

You can configure PVLAN trunk secondary/promiscuous in the of switchport mode. This is supported for both physical interface and portchannel.

Port Type Changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

Access port to trunk port

When you change a Layer 2 interface from an access port to a trunk port, the device deletes all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN. The sticky MAC addresses remain in same VLAN if the VLAN exists. Otherwise, the MAC addresses move to the native VLAN of the trunk port.

Trunk port to access port

When you change a Layer 2 interface from a trunk port to an access port, the device drops all secure addresses learned by the dynamic method. It also moves all addresses learned by the sticky method on the native trunk VLAN to the access VLAN. The device drops secure addresses learned by the sticky method if they are not on the native trunk VLAN.

Switched port to routed port

When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

Routed port to switched port

When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

The static secure addresses that are configured per access or trunk VLAN on an interface are not retained during the following events:

- Changing global VLAN mode of the active VLANs on an interface between classical Ethernet and fabric path interfaces
- Changing switchport mode access or trunk to private VLAN or vice versa

802.1X and Port Security

You can configure port security and 802.1X on the same interfaces. Port security secures the MAC addresses that 802.1X authenticates. 802.1X processes packets before port security processes them, so when you enable both on an interface, 802.1X is already preventing inbound traffic on the interface from unknown MAC addresses.

When you enable 802.1X and port security on the same interface, port security continues to learn MAC addresses by the sticky or dynamic method, as configured. Additionally, depending on whether you enable 802.1X in single-host mode or multiple-host mode, one of the following occurs:

Single host mode

Port security learns the MAC address of the authenticated host.

Multiple host mode

Port security drops any MAC addresses learned for this interface by the dynamic method and learns the MAC address of the first host authenticated by 802.1X.

If a MAC address that 802.1X passes to port security would violate the applicable maximum number of secure MAC addresses, the device sends an authentication failure message to the host.

The device treats MAC addresses authenticated by 802.1X as though they were learned by the dynamic method, even if port security previously learned the address by the sticky or static methods. If you attempt to delete a secure MAC address that has been authenticated by 802.1X, the address remains secure.

If the MAC address of an authenticated host is secured by the sticky or static method, the device treats the address as if it were learned by the dynamic method, and you cannot delete the MAC address manually.

Port security integrates with 802.1X to reauthenticate hosts when the authenticated and secure MAC address of the host reaches its port security age limit. The device behaves differently depending upon the type of aging, as follows:

Absolute

Port security notifies 802.1X and the device attempts to reauthenticate the host. The result of reauthentication determines whether the address remains secure. If reauthentication succeeds, the device restarts the aging timer on the secure address; otherwise, the device drops the address from the list of secure addressees for the interface.

Inactivity

Port security drops the secure address from the list of secure addresses for the interface and notifies 802.1X. The device attempts to reauthenticate the host. If reauthentication succeeds, port security secures the address again.

Licensing Requirements for Port Security

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Port security requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS device images and is provided at no extra charge to you.

Prerequisites for Port Security

Guidelines and Limitations for Port Security

When configuring port security, follow these guidelines:

- Port security is supported on PVLAN ports.
- Port security does not support switched port analyzer (SPAN) destination ports.
- Port security does not depend upon other features.
- If any member link in a port-channel is in the pre-provisioned state, that is, the module is offline, then the port security feature cannot be disabled on the port-channel.

Guidelines and Limitations for Port Security on vPCs

In addition to the guidelines and limitations for port security, there are additional guidelines and limitations for port security on vPCs. When configuring port security on vPCs, follow these guidelines:

- You must enabled port security globally on both vPC peers in a vPC domain.
- You must enable port security on the vPC interfaces of both vPC peers.
- You must configure a static secure MAC address on the primary vPC peer. This MAC address is synchronized with the secondary vPC peer. You can also configure a static secure MAC address on the secondary peer. This MAC address appears in the secondary vPC configuration, but does not take affect.
- All learned MAC addresses are synchronized between vPC peers.
- Both vPC peers can be configured with either the dynamic or sticky MAC address learning method. However, we recommend that both vPC peers be configured for the same method.
-

- Dynamic MAC addresses are dropped only after the age limit is reached on both vPC peers.
- You set the maximum number of secure MAC addresses on the primary vPC switch. The primary vPC switch does the count validation, even if a maximum number of secure MAC addresses is set on the secondary switch.
- You configure the violation action on the primary vPC. So, whenever a security violation is triggered, the security action defined on the primary vPC switch occurs.
- Port security is enabled on a vPC interface when the port security feature is enabled on both vPC peers and port security is enabled on both vPC interfaces of the vPC peers. You can use the **config sync** command to verify that the configuration is correct.
- While a switch undergoes an in-service software upgrade (ISSU), port security operations are stopped on its peer switch. The peer switch does not learn any new MAC addresses, and MAC moves occurring during this operation are ignored. When the ISSU is complete, the peer switch is notified and normal port security functionality resumes.
- ISSU to higher versions is supported; however ISSU to lower versions is not supported.

Configuring Port Security

Enabling or Disabling Port Security Globally

You can enable or disable port security globally on a device. By default, port security is disabled globally.

When you disable port security, all port security configuration on the interface is ineffective. When you disable port security globally, all port security configuration is lost.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature port-security** **Example:** `switch(config)# feature port-security`	Enables port security globally. The **no** option disables port security globally.
Step 3	**show port-security** **Example:** `switch(config)# show port-security`	Displays the status of port security.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

You can enable port-security on a port-channel in the following ways:

- Bundle member links into a port-channel by using the **channel-group** command and then enable port-security on the port-channel.
- Create port-channel and configure port security. Configure port security on member links and then bundle member links by using the **channel-group** command. In case of pre-provisioned member links, you can bundle them to the port-channel after the module is online.

Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot/port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the Ethernet or port-channel interface that you want to configure with port security.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

	Command or Action	Purpose
Step 4	[**no**] **switchport port-security** **Example:** `switch(config-if)# switchport port-security`	Enables port security on the interface. The **no** option disables port security on the interface.
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Sticky MAC Address Learning

You can disable or enable sticky MAC address learning on an interface. If you disable sticky learning, the device returns to dynamic MAC address learning on the interface, which is the default learning method.

By default, sticky MAC address learning is disabled.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with sticky MAC address learning.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

	Command or Action	Purpose
Step 4	[**no**] **switchport port-security mac-address sticky** **Example:** `switch(config-if)# switchport port-security mac-address sticky`	Enables sticky MAC address learning on the interface. The **no** option disables sticky MAC address learning.
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Adding a Static Secure MAC Address on an Interface

You can add a static secure MAC address on a Layer 2 interface.

Note If the MAC address is a secure MAC address on any interface, you cannot add it as a static secure MAC address to another interface until you remove it from the interface on which it is already a secure MAC address.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

You must have enabled port security globally.

Verify that the interface maximum has not been reached for secure MAC addresses. If needed, you can remove a secure MAC address or you can change the maximum number of addresses on the interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **interface ethernet** *slot/port* • **interface port-channel** *channel-number*	Enters interface configuration mode for the interface that you specify.

	Command or Action	Purpose

	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`	

Step 3	[**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*] **Example:** `switch(config-if)# switchport port-security mac-address 0019.D2D0.00AE`	Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing a Static Secure MAC Address on an Interface

You can remove a static secure MAC address on a Layer 2 interface.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a static secure MAC address.

Step 3	**no switchport port-security mac-address** *address* **Example:** `switch(config-if)# no switchport port-security mac-address 0019.D2D0.00AE`	Removes the static secure MAC address from port security on the current interface.

	Command or Action	Purpose
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing a Sticky Secure MAC Address

You can remove a sticky secure MAC addresses, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a sticky secure MAC address.

Step 3	**no switchport port-security mac-address sticky** **Example:** `switch(config-if)# no switchport port-security mac-address sticky`	Disables sticky MAC address learning on the interface, which converts any sticky secure MAC addresses on the interface to dynamic secure MAC addresses.
Step 4	**clear port-security dynamic address** *address* **Example:** `switch(config-if)# clear port-security dynamic address 0019.D2D0.02GD`	Removes the dynamic secure MAC address that you specify.

	Command or Action	Purpose
Step 5	(Optional) **show port-security address interface** {**ethernet** *slot*/*port* | **port-channel** *channel-number*} **Example:** `switch(config)# show port-security address`	Displays secure MAC addresses. The address that you removed should not appear.
Step 6	(Optional) **switchport port-security mac-address sticky** **Example:** `switch(config-if)# switchport port-security mac-address sticky`	Enables sticky MAC address learning again on the interface.

Removing a Dynamic Secure MAC Address

You can remove dynamically learned, secure MAC addresses.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**clear port-security dynamic** {**interface ethernet** *slot*/*port* | **address** *address*} [**vlan** *vlan-ID*] **Example:** `switch(config)# clear port-security dynamic interface ethernet 2/1`	Removes dynamically learned, secure MAC addresses, as specified. If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify. If you use the **address** keyword, you remove the single, dynamically learned address that you specify. Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.
Step 3	**show port-security address** **Example:** `switch(config)# show port-security address`	Displays secure MAC addresses.

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

Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of address is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Note When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the device rejects the command. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot* is the interface that you want to configure with the maximum number of MAC addresses.

Step 3	[**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*] **Example:** `switch(config-if)# switchport port-security maximum 425`	Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 1025. The **no** option resets the maximum number of MAC addresses to the default, which is 1.

	Command or Action	Purpose
		If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an Address Aging Type and Time

You can configure the MAC address aging type and the length of time that the device uses to determine when MAC addresses learned by the dynamic method have reached their age limit.

Absolute aging is the default aging type.

By default, the aging time is 0 minutes, which disables aging.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with the MAC aging type and time.

Step 3	[**no**] **switchport port-security aging type** {**absolute** | **inactivity**} **Example:** `switch(config-if)# switchport port-security aging type inactivity`	Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging.

	Command or Action	Purpose
		Note F1 series modules do not support the **inactivity** aging type.
Step 4	[**no**] **switchport port-security aging time** *minutes* **Example:** `switch(config-if)# switchport port-security aging time 120`	Configures the number of minutes that a dynamically learned MAC address must age before the device drops the address. The maximum valid *minutes* is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Security Violation Action

You can configure the action that the device takes if a security violation occurs. The violation action is configurable on each interface that you enable with port security.

The default security action is to shut down the port on which the security violation occurs.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with a security violation action.

	Command or Action	Purpose
Step 3	[**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**} **Example:** `switch(config-if)# switchport port-security violation restrict`	Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Port Security Configuration

To display the port security configuration information, perform one of the following tasks. For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference*.

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status of the device.
show port-security interface	Displays the port security status of a specific interface.
show port-security address	Displays secure MAC addresses.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses. For detailed information about the fields in the output from this command, see the

Configuration Example for Port Security

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

Configuration Example of Port Security in a vPC Domain

The following example shows how to enable and configure port security on vPC peers in a vPC domain. The first switch is the primary vPC peer and the second switch is the secondary vPC peer. It is assumed that domain 103 has already been created.

```
primary_switch(config)# feature port-security
primary_switch(config-if)# int e1/1
primary_switch(config-if)# switchport port-security
primary_switch(config-if)# switchport port-security max 1025
primary_switch(config-if)# switchport port-security violation restrict
primary_switch(config-if)# switchport port-security aging time 4
primary_switch(config-if)# switchport port-security aging type absolute
primary_switch(config-if)# switchport port-security mac sticky
primary_switch(config-if)# switchport port-security mac-address 0.0.1 vlan 101
primary_switch(config-if)# switchport port-security mac-address 0.0.2 vlan 101
primary_switch(config-if)# copy running-config startup-config

secondary_switch(config)# int e103/1/1
secondary_switch(config-if)# switchport port-security
secondary_switch(config-if)# copy running-config startup-config
```

Default Settings for Port Security

This table lists the default settings for port security parameters.

Table 17: Default Port Security Parameters

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown
Aging type	Absolute
Aging time	0

Additional References for Port Security

Related Documents

Standards

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

MIBs

Cisco NX-OS provides read-only SNMP support for port security.

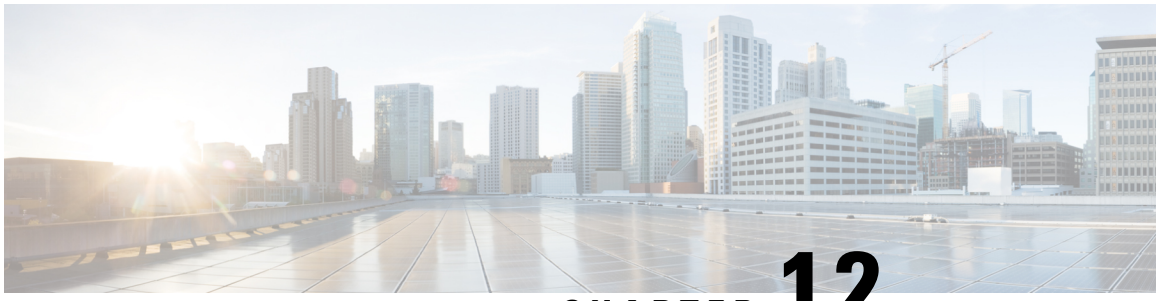
MIBs	MIBs Link
• CISCO-PORT-SECURITY-MIB **Note** Traps are supported for notification of secure MAC address violations.	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for Port Security

This table lists the release history for this feature.

Table 18: Feature History for Port Security

Feature Name	Releases	Feature Information	
Port security	4.2(1)		



CHAPTER 12

Configuring DHCP Snooping

This chapter contains the following sections:

Information About DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

Feature Enabled and Globally Enabled

When you are configuring DHCP snooping, it is important that you understand the difference between enabling the DHCP snooping feature and globally enabling DHCP snooping.

Feature Enablement

The DHCP snooping feature is disabled by default. When the DHCP snooping feature is disabled, you cannot configure it or any of the features that depend on DHCP snooping. The commands to configure DHCP snooping and its dependent features are unavailable when DHCP snooping is disabled.

When you enable the DHCP snooping feature, the switch begins building and maintaining the DHCP snooping binding database. Features dependent on the DHCP snooping binding database can now make use of it and can therefore also be configured.

Enabling the DHCP snooping feature does not globally enable it. You must separately enable DHCP snooping globally.

Disabling the DHCP snooping feature removes all DHCP snooping configuration from the switch. If you want to disable DHCP snooping and preserve the configuration, globally disable DHCP snooping but do not disable the DHCP snooping feature.

Global Enablement

After DHCP snooping is enabled, DHCP snooping is globally disabled by default. Global enablement is a second level of enablement that allows you to have separate control of whether the switch is actively performing DHCP snooping that is independent from enabling the DHCP snooping binding database.

When you globally enable DHCP snooping, on each untrusted interface of VLANs that have DHCP snooping enabled, the switch begins validating DHCP messages that are received and used the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

When you globally disable DHCP snooping, the switch stops validating DHCP messages and validating subsequent requests from untrusted hosts. It also removes the DHCP snooping binding database. Globally disabling DHCP snooping does not remove any DHCP snooping configuration or the configuration of other features that are dependent upon the DHCP snooping feature.

Trusted and Untrusted Sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source might initiate traffic attacks or other hostile actions. To prevent such attacks, DHCP snooping filters messages from untrusted sources.

In an enterprise network, a trusted source is a switch that is under your administrative control. These switches include the switches, routers, and servers in the network. Any switch beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any switch that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In a Cisco Nexus device, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to switches (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, you must connect all DHCP servers to the switch through trusted interfaces.

DHCP Snooping Binding Database

Using information extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts that are connected through trusted interfaces.

Note The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

DHCP snooping updates the database when the switch receives specific DHCP messages. For example, the feature adds an entry to the database when the switch receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the switch receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

Information About the DHCPv6 Relay Agent

DHCPv6 Relay Agent

You can configure the device to run a DHCPv6 relay agent, which forwards DHCPv6 packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCPv6 messages and then generate a new DHCPv6 message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCPv6 packet) and forwards it to the DHCPv6 server.

VRF Support for the DHCPv6 Relay Agent

You can configure the DHCPv6 relay agent to forward DHCPv6 broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCPv6 servers in a different VRF. By using a single DHCPv6 server to provide DHCPv6 support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF.

Information About the Lightweight DHCPv6 Relay Agent

Lightweight DHCPv6 Relay Agent

A variety of different link-layer network topologies exist for the aggregation of IPv6 nodes into one or more routers. In Layer 2 aggregation networks (IEEE 802.1D bridging or similar) that have many nodes on a single link, a DHCP Version 6 (DHCPv6) server or DHCP relay agent normally does not recognize how a DHCP client is attached to a network. From Cisco NX-OS Release , you can configure the interface of a device to run Lightweight DHCPv6 Relay Agent (LDRA), which forwards DHCPv6 messages between clients and servers.

The LDRA feature is used to insert relay agent options in DHCPv6 message exchanges primarily to identify client-facing interfaces. LDRA resides on the same IPv6 link as the client and a DHCPv6 relay agent or server.

LDRA for VLANs and Interfaces

You can configure LDRA on VLANs and interfaces. LDRA is not enabled by default. To enable LDRA, it should be enabled globally and at the interface level. You should configure the interfaces as client-facing trusted, client-facing untrusted, or server-facing. All client-facing interfaces must be configured as trusted or untrusted. By default, all the client-facing interfaces in LDRA are configured as untrusted. When a client-facing interface is deemed untrusted, LDRA will discard messages of type RELAY-FORWARD, which are received from the client-facing interface.

The LDRA configuration on a VLAN should be configured as client-facing trusted or client-facing untrusted. When you configure LDRA functionality on a VLAN, the functionality is configured on all the ports or interfaces within the VLAN. However, if you configure an interface in a VLAN as client-facing untrusted, and configure the VLAN as client-facing trusted, the configuration of an interface takes precedence over the configuration of a VLAN. At least one interface in a VLAN should be configured as server-facing interface.

Guidelines and Limitations for Lightweight DHCPv6 Relay Agent

- Access nodes implementing LDRA do not support IPv6 control or routing.
- An interface or port cannot be configured as both client facing and server facing at the same time.
- To support virtual port channel, LDRA configuration should be symmetric on the vPC peers.
- LDRA supports Cisco Fabricpath.

Guidelines and Limitations for DHCP Snooping

Consider the following guidelines and limitations when configuring DHCP snooping:

- The DHCP snooping database can store 2000 bindings.
- DHCP snooping is not active until you enable the feature, enable DHCP snooping globally, and enable DHCP snooping on at least one VLAN.

- Before globally enabling DHCP snooping on the switch, make sure that the switches that act as the DHCP server and the DHCP relay agent are configured and enabled.
- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.
- DHCP snooping does not work with DHCP relay configured on the same nexus device.
- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.

Default Settings for DHCP Snooping

This table lists the default settings for DHCP snooping parameters.

Table 19: Default DHCP Snooping Parameters

Parameters	Default
DHCP snooping feature	Disabled
DHCP snooping globally enabled	No
DHCP snooping VLAN	None
DHCP snooping Option 82 support	Disabled
DHCP snooping trust	Untrusted
VRF support for the DHCP relay agent	Disabled
VRF support for the DHCPv6 relay agent	Disabled
DHCP relay agent	Disabled
DHCPv6 relay agent	Disabled
DHCPv6 relay option type cisco	Disabled

Configuring DHCP Snooping

Minimum DHCP Snooping Configuration

1. Enable the DHCP snooping feature.
2.

Procedure

	Command or Action	Purpose
Step 1	Enable the DHCP snooping feature.	When the DHCP snooping feature is disabled, you cannot configure DHCP snooping. For details, see Enabling or Disabling the DHCP Snooping Feature, on page 194.
Step 2	Enable DHCP snooping globally.	For details, see Enabling or Disabling DHCP Snooping Globally, on page 195.
Step 3	Enable DHCP snooping on at least one VLAN.	By default, DHCP snooping is disabled on all VLANs. For details, see Enabling or Disabling DHCP Snooping on a VLAN, on page 195.
Step 4	Ensure that the DHCP server is connected to the switch using a trusted interface.	For details, see Configuring an Interface as Trusted or Untrusted, on page 197.

Enabling or Disabling the DHCP Snooping Feature

You can enable or disable the DHCP snooping feature on the switch. By default, DHCP snooping is disabled.

Before you begin

If you disable the DHCP snooping feature, all DHCP snooping configuration is lost. If you want to turn off DHCP snooping and preserve the DHCP snooping configuration, disable DHCP globally.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature dhcp** **Example:** `switch(config)# feature dhcp`	Enables the DHCP snooping feature. The **no** option disables the DHCP snooping feature and erases all DHCP snooping configuration.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

Enabling or Disabling DHCP Snooping Globally

You can enable or disable the DHCP snooping globally on the switch. Globally disabling DHCP snooping stops the switch from performing any DHCP snooping but preserves DCHP snooping configuration.

Before you begin

Ensure that you have enabled the DHCP snooping feature. By default, DHCP snooping is globally disabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** option disables DHCP snooping.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs.

Before you begin

By default, DHCP snooping is disabled on all VLANs.

Ensure that DHCP snooping is enabled.

Note If a VACL is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:** `switch(config)# ip dhcp snooping vlan 100,200,250-252`	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP snooping on the VLANs specified.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Strict DHCP Packet Validation

You can enable or disable the strict validation of DHCP packets by the DHCP snooping feature. By default, strict validation of DHCP packets is disabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp packet strict-validation** **Example:** `switch(config)# ip dhcp packet strict-validation`	Enables the strict validation of DHCP packets by the DHCP snooping feature. The **no** option disables strict DHCP packet validation.

	Command or Action	Purpose
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an Interface as Trusted or Untrusted

You can configure whether an interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

By default, all interfaces are untrusted.

Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **interface ethernet** *port*/*slot*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		• Enters interface configuration mode, where *port* / *slot* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.
		• Enters interface configuration mode, where *port* / *slot* is the Layer 2 port-channel interface that you want to configure as trusted or untrusted for DHCP snooping.

Step 3	[**no**] **ip dhcp snooping trust** **Example:** `switch(config-if)# ip dhcp snooping trust`	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.

	Command or Action	Purpose
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling the DHCP Relay Agent

You can enable or disable the DHCP relay agent. By default, the DHCP relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a DHCP Static Binding

You can create a static DHCP source binding to a Layer 2 interface.

Before you begin

Ensure that you have enabled the DHCP snooping feature.

Procedure

	Command or Action	Purpose

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

Step 2	**ip source binding** *IP-address MAC-address* **vlan** *vlan-id* {**interface ethernet** *slot/port* | **port-channel** *channel-no*} **Example:** `switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3`	Binds the static source address to the Layer 2 Ethernet interface.
Step 3	(Optional) **show ip dhcp snooping binding** **Example:** `switch(config)#  ip dhcp snooping binding`	Shows the DHCP snooping static and dynamic bindings.
Step 4	(Optional) **show ip dhcp snooping binding dynamic** **Example:** `switch(config)#  ip dhcp snooping binding dynamic`	Shows the DHCP snooping dynamic bindings.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

The following example shows how to create a static IP source entry associated with VLAN 100 on Ethernet interface 2/3:

```
switch# configure terminal
switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3
switch(config)#
```

Configuring DHCPv6

Enabling or Disabling the DHCPv6 Relay Agent

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ipv6 dhcp relay** **Example:** `switch(config)# ipv6 dhcp relay`	Enables the DHCPv6 relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling VRF Support for the DHCPv6 Relay Agent

You can configure the device to support the relaying of DHCPv6 requests that arrive on an interface in one VRF to a DHCPv6 server in a different VRF.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ipv6 dhcp relay option vpn** **Example:** `switch(config)# ipv6 dhcp relay option vpn`	Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior.
Step 3	[**no**] **ipv6 dhcp relay option type cisco** **Example:** `switch(config)# ipv6 dhcp relay option type cisco`	Causes the DHCPv6 relay agent to insert virtual subnet selection (VSS) details as part of the vendor-specific option. The **no** option causes the DHCPv6 relay agent to insert VSS details as part of the VSS option (68), which is defined in RFC-6607. This command is useful when you want to use DHCPv6 servers that do not support RFC-6607 but allocate IPv6 addresses based on the client VRF name.
Step 4	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the DHCPv6 Relay Source Interface

You can configure the source interface for the DHCPv6 relay agent. By default, the DHCPv6 relay agent uses the relay agent address as the source address of the outgoing packet. Configuring the source interface enables you to use a more stable address (such as the loopback interface address) as the source address of relayed messages.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ipv6 dhcp relay source-interface** *interface* **Example:** `switch(config)# ipv6 dhcp relay source-interface loopback 2`	Configures the source interface for the DHCPv6 relay agent. **Note** The DHCPv6 relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Lightweight DHCPv6 Relay Agent

Configuring Lightweight DHCPv6 Relay Agent for an Interface

Perform this task to configure Lightweight DHCPv6 Relay Agent (LDRA) for an interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**Example:**	Enables the LDRA functionality globally.
Step 3	**interface** *slot*/*port* **Example:** `switch(config)# interface ethernet 0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**switchport** **Example:** `switch(config-if)# switchport`	Switches an interface that is in Layer 3 mode to Layer 2 mode for Layer 2 configuration.
Step 5	**[no] ipv6 dhcp-ldra {client-facing-trusted | client-facing-untrusted | client-facing-disable | server-facing}** **Example:** `switch(config-if)# ipv6 dhcp-ldra server-facing`	Enables LDRA functionality on a specified interface or port. The **no** option disables the LDRA functionality. **Note** The **client-facing-trusted** specifies client-facing interfaces or ports as trusted. The trusted port allows the DHCPv6 packets and they are encapsulated as per LDRA options. The **client-facing-untrusted** specifies client-facing interfaces or ports as untrusted. The untrusted ports perform LDRA functionality, but drop only the relay forward packets received on it. The **client-facing-disable** keyword disables LDRA functionality on an interface or port. Disabled port performs the Layer-2 forwarding of DHCPv6 packets. The **server-facing** keyword specifies an interface or port as server facing. Server facing port allows the reply packets from server.

Configuring Lightweight DHCPv6 Relay Agent for a VLAN

Perform this task to configure Lightweight DHCPv6 Relay Agent (LDRA) for a VLAN.

Before you begin

Ensure that the VLAN is not assigned an IP address.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**Example:**	Enables the LDRA functionality globally.
Step 3	**[no] ipv6 dhcp-ldra attach–policy vlan** *vlan-id* **{client-facing-trusted | client-facing-untrusted}** **Example:** `switch(config)# ipv6 dhcp-ldra attach-policy vlan 25 client-facing-trusted`	Enables LDRA functionality on the specified VLAN. The **no** option disables the LDRA functionality. **Note** The **client-facing-trusted** keyword configures all the ports or interfaces associated with the VLAN as client-facing, trusted ports. The **client-facing-untrusted** keyword configures all the ports or interfaces associated with the VLAN as client-facing, untrusted ports.

Verifying the DHCP Snooping Configuration

To display DHCP snooping configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the System Management Configuration Guide for your Cisco Nexus device.

Command	Purpose
show running-config dhcp	Displays the DHCP snooping configuration.
show ip dhcp relay	Displays the DHCP relay configuration.
show ipv6 dhcp relay [**interface** *interface*]	Displays the DHCPv6 relay global or interface-level configuration.
show ip dhcp snooping	Displays general information about DHCP snooping.

Displaying DHCP Bindings

Use the **show ip dhcp snooping binding** command to display the DHCP static and dynamic binding table. Use the **show ip dhcp snooping binding dynamic** to display the DHCP dynamic binding table.

For detailed information about the fields in the output from this command, see the *System Management Configuration Guide* for your Cisco Nexus device.

This example shows how to create a static DHCP binding and then verify the binding using the **show ip dhcp snooping binding** command.

```
switch# configuration terminal
switch(config)# ip source binding 10.20.30.40 0000.1111.2222 vlan 400 interface port-channel
 500

switch(config)# show ip dhcp snooping binding
MacAddress          IpAddress        LeaseSec  Type        VLAN  Interface
------------------  ---------------  --------  ----------  ----  -------------
00:00:11:11:22:22  10.20.30.40      infinite  static      400   port-channel500
```

Displaying and Clearing LDRA Information

To clear the DHCPv6 LDRA-specific statistics, use the **clear ipv6 dhcp-ldra statistics** command.

Clearing the DHCP Snooping Binding Database

You can remove entries from the DHCP snooping binding database, including a single entry, all entries associated with an interface, or all entries in the database.

Before you begin

Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **clear ip dhcp snooping binding** **Example:** `switch# clear ip dhcp snooping binding`	Clears all entries from the DHCP snooping binding database.
Step 2	(Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*] **Example:** `switch# clear ip dhcp snooping binding interface ethernet 1/4`	Clears entries associated with a specific Ethernet interface from the DHCP snooping binding database.

	Command or Action	Purpose
Step 3	(Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*] **Example:** `switch# clear ip dhcp snooping binding interface port-channel 72`	Clears entries associated with a specific port-channel interface from the DHCP snooping binding database.
Step 4	(Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] } **Example:** `switch# clear ip dhcp snooping binding vlan 23 mac 0060.3aeb.54f0 ip 10.34.54.9 interface ethernet 2/11`	Clears a single, specific entry from the DHCP snooping binding database.
Step 5	(Optional) **show ip dhcp snooping binding** **Example:** `switch# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Clearing DHCP Relay Statistics

Use the **clear ip dhcp relay statistics** command to clear the global DHCP relay statistics.

Use the **clear ip dhcp relay statistics interface** *interface* command to clear the DHCP relay statistics for a particular interface.

Use the **clear ip dhcp relay statistics interface** *interface* **serverip** *ip-address* [**use-vrf** *vrf-name*] command to clear the DHCP relay statistics at the server level for a particular interface.

Clearing DHCPv6 Relay Statistics

Use the **clear ipv6 dhcp relay statistics** command to clear the global DHCPv6 relay statistics.

Use the **clear ipv6 dhcp relay statistics interface** *interface* command to clear the DHCPv6 relay statistics for a particular interface.

Use the **clear ipv6 dhcp relay statistics interface** *interface* **server-ip** *ip-address* [**use-vrf** *vrf-name*] command to clear the DHCPv6 relay statistics at the server level for a particular interface.

Monitoring DHCP

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.

Use the **show ip dhcp relay statistics** [**interface** *interface* [**serverip** *ip-address* [**use-vrf** *vrf-name*]]] command to monitor DHCP relay statistics at the global, server, or interface level.

Use the (Optional) **show ip dhcp snooping statistics vlan** [*vlan-id*] **interface** [**ethernet**|*port-channel*][*id*] command to know the exact statistics about snooping statistics per interface under a vlan.

Configuration Examples for DHCP Snooping

The following example shows how to enable DHCP snooping on two VLANs, with Option 82 support enabled and Ethernet interface 2/5 trusted because the DHCP server is connected to that interface:

```
feature dhcp
ip dhcp snooping
ip dhcp snooping info option

interface Ethernet 2/5
  ip dhcp snooping trust
ip dhcp snooping vlan 1
ip dhcp snooping vlan 50
```

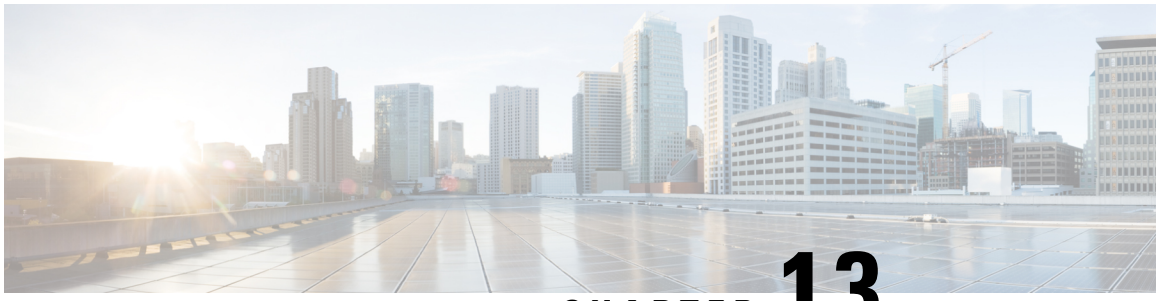
Configuration Examples for LDRA

Configuring LDRA for an Interface

The following example shows how to enable LDRA and configure interface Ethernet 1/1 as client-facing and trusted:

Configuring LDRA for a VLAN

The following example shows how to enable LDRA and configure VLAN with VLAN ID 25 as client-facing and trusted:



CHAPTER 13

Configuring Control Plane Policing

This chapter contains the following sections:

Information About CoPP

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

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic destined to any of the IP addresses of the router or Layer 3 switch. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

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

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

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

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

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

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

Examples of DoS attacks are as follows:

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

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

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

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

Control Plane Protection

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

Control Plane Packet Types

Different types of packets can reach the control plane:

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

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

Redirected packets
Packets that are redirected to the supervisor module. Features such as Dynamic Host Configuration Protocol (DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.

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

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

Classification for CoPP

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

Rate Controlling Mechanisms

Once the packets are classified, the Cisco NX-OS device has two different mechanisms to control the rate at which packets arrive at the supervisor module: policing and rate limiting.

Using hardware policers, you can define separate actions for traffic that conforms to or violates certain conditions. These 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.

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

CoPP Class Maps

The following table shows the available class maps and their configurations.

Table 20: Class Map Configurations and Descriptions

Class Map	Configuration	Description
class-map type control-plane match-any copp-system-class-arp	match protocol arp match protocol nd	Class matches all ARP packets. Class matches all ARP packets and ND (NA, NS, RA, and RS) packets.
class-map type control-plane match-any copp-system-class-bgp	match protocol bgp	Class matches all BGP packets.
class-map type control-plane match-any copp-system-class-bridging	match protocol bridging	Class matches all STP and RSTP frames.
class-map type control-plane match-any copp-system-class-cdp	match protocol cdp	Class matches all CDP frames.
class-map type control-plane match-any copp-system-class-default	match protocol default	Class matches all frames. Used for the default policer.
class-map type control-plane match-any copp-system-class-dhcp	match protocol dhcp	Class matches all IPv4 DHCP packets Class matches all both IPv4 DHCP packets.
class-map type control-plane match-any copp-system-class-eigrp	match protocol eigrp match protocol eigrp6	Class matches all IPv4 EIGRP packets. Class matches both IPv4 and IPv6 EIGRP packets.
class-map type control-plane match-any copp-system-class-exception	match protocol exception	Class matches all IP packets that are treated as exception packets (except TTL exception, IP Fragment exception and Same Interface exception packets) for IP routing purposes, such as packets with a Martian destination address or with an MTU failure.
class-map type control-plane match-any copp-system-class-excp-ip-frag	match protocol ip_frag	Class matches all IP packets that are fragments. (These packets are treated as exception packets from an IP routing perspective).
class-map type control-plane match-any copp-system-class-excp-same-if	match protocol same-if	Class matches all IP packets that are treated as exception packets for IP routing. The packets are matched because they are received from the interface where their destination is supposed to be.

Class Map	Configuration	Description
class-map type control-plane match-any copp-system-class-excp-ttl	match protocol ttl	Class matches all packets that are treated as TTL exception packets (when TTL is 0) from a IP routing perspective.
class-map type control-plane match-any copp-system-class-fip	match protocol fip	Class matches all packets belonging to the FCoE Initialization Protocol.
class-map type control-plane match-any copp-system-class-glean	match protocol glean	
class-map type control-plane match-any copp-system-class-hsrp-vrrp	match protocol hsrp_vrrp match protocol hsrp6	Class matches HSRP and VRRP packets. Class matches IPv4 HSRP, VRRP and IPv6 HSRP packets
class-map type control-plane match-any copp-system-class-icmp-echo	match protocol icmp_echo	Class matches all ICMP Echo (Ping) packets.
class-map type control-plane match-any copp-system-class-igmp	match protocol igmp	Class matches all IGMP packets.
class-map type control-plane match-any copp-system-class-isis	match protocol isis_dce	
class-map type control-plane match-any copp-system-class-l3dest-miss	match protocol unicast	Class matches all unicast routed packets that did not find a destination in the FIB.
class-map type control-plane match-any copp-system-class-lacp	match protocol lacp	Class matches all Link Aggregation Control Protocol (LACP) frames.
class-map type control-plane match-any copp-system-class-lldp	match protocol lldp_dcx	Class matches all LLDP frames.
class-map type control-plane match-any copp-system-class-mcast-last-hop	match protocol mcast_last_hop	Class matches all IP multicast last hop packets.
class-map type control-plane match-any copp-system-class-mcast-miss	match protocol multicast	Class matches all IP multicast frames that could not be routed because they did not have an entry in the FIB.
class-map type control-plane match-any copp-system-class-mgmt	match protocol mgmt	Class matches all management-related frames, such as SNMP, HTTP, NTP, Telnet, and SSH.
class-map type control-plane match-any copp-system-class-msdp	match protocol msdp	Class matches MSDP packets.

Class Map	Configuration	Description
class-map type control-plane match-any copp-system-class-ospf	match protocol ospf match protocol ospfv3	Class matches OSPF and OSPFv3 Protocol packets.
class-map type control-plane match-any copp-system-class-pim-hello	match protocol pim	Class matches all PIM Hello packets.
class-map type control-plane match-any copp-system-class-pim-register	match protocol reg	Class matches all PIM Register packets.
class-map type control-plane match-any copp-system-class-rip	match protocol rip	Class matches all RIP packets.
class-map type control-plane match-any copp-system-class-rpf-fail	match protocol rpf_fail	Class matches all RPF failure packets.
class-map type control-plane match-any copp-system-class-udld	match protocol udld	Class matches all UDLD frames.

CoPP Policy Templates

When you bring up your Cisco NX-OS device for the first time, the Cisco NX-OS software installs the default copp-system-policy to protect the supervisor module from DoS attacks. You can choose the CoPP policy template for your deployment scenario by specifying CoPP policy options from the initial setup utility:

- Default CoPP Policy (copp-system-policy-default)
- Scaled Layer 2 CoPP Policy (copp-system-policy-scaled-l2)
- Scaled Layer 3 CoPP Policy (copp-system-policy-scaled-l3)
- Customized CoPP Policy (copp-system-policy-customized)

If you do not select an option or choose not to execute the setup utility, the Cisco NX-OS software applies the Default policing. Cisco recommends starting with the default policy and later modifying the CoPP policies as required.

The default copp-system-policy-default policy has optimized values suitable for basic device operations.

You can change which CoPP policy is used by using the **service-policy input** *policy-name* command in the control plane configuration mode.

Default CoPP Policy

The copp-system-policy-default policy is applied to the switch by default. It has the classes with policer rates that should suit most network installations. You cannot modify this policy or the class maps associated with it. In addition, you cannot modify the class map configurations in this policy.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy-default
    class copp-system-class-igmp
      police cir 1024 kbps bc 65535 bytes
    class copp-system-class-pim-hello
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bridging
      police cir 20000 kbps bc 4800000 bytes
    class copp-system-class-arp
      police cir 1024 kbps bc 3600000 bytes
    class copp-system-class-dhcp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-mgmt
      police cir 12000 kbps bc 4800000 bytes
    class copp-system-class-lacp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-lldp
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-udld
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-isis
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-msdp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-cdp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-fip
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bgp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-eigrp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-exception
      police cir 64 kbps bc 4800000 bytes
    class copp-system-class-glean
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-hsrp-vrrp
      police cir 1024 kbps bc 256000 bytes
    class copp-system-class-icmp-echo
      police cir 64 kbps bc 3600000 bytes
    class copp-system-class-ospf
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-pim-register
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-rip
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-l3dest-miss
      police cir 64 kbps bc 256000 bytes
    class copp-system-class-mcast-miss
      police cir 256 kbps bc 3200000 bytes
    class copp-system-class-excp-ip-frag
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-excp-same-if
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-excp-ttl
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-default
      police cir 512 kbps bc 6400000 bytes
```

Scaled Layer 2 CoPP Policy

The copp-system-policy-scaled policy has most classes with policer rates that are same as the default policy. However, it has higher policer rates for IGMP and ISIS. You cannot modify this policy or the class maps associated with it. In addition, you cannot modify the class map configurations in this policy.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy-scaled-l2
    class copp-system-class-igmp
      police cir 4096 kbps bc 264000 bytes
    class copp-system-class-pim-hello
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bridging
      police cir 20000 kbps bc 4800000 bytes
    class copp-system-class-arp
      police cir 1024 kbps bc 3600000 bytes
    class copp-system-class-dhcp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-mgmt
      police cir 12000 kbps bc 4800000 bytes
    class copp-system-class-lacp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-lldp
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-udld
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-isis
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-msdp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-cdp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-fip
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bgp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-eigrp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-exception
      police cir 64 kbps bc 4800000 bytes
    class copp-system-class-glean
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-hsrp-vrrp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-icmp-echo
      police cir 64 kbps bc 3600000 bytes
    class copp-system-class-ospf
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-pim-register
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-rip
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-l3dest-miss
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-mcast-miss
      police cir 256 kbps bc 3200000 bytes
    class copp-system-class-excp-ip-frag
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-excp-same-if
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-excp-ttl
```

```
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-default
      police cir 512 kbps bc 6400000 bytes
```

Scaled Layer 3 CoPP Policy

The copp-system-policy-scaled-l3 policy has most classes with policer rates that are same as the default policy. However, it has higher policer rates for IGMP, ICMP Echo, ISIS, Mcast-miss, and Glean related classes. You cannot modify this policy or the class maps associated with it. In addition, you cannot modify the class map configurations in this policy.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy-scaled-l3
    class copp-system-class-igmp
      police cir 4096 kbps bc 264000 bytes
    class copp-system-class-pim-hello
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bridging
      police cir 20000 kbps bc 4800000 bytes
    class copp-system-class-arp
      police cir 4000 kbps bc 3600000 bytes
    class copp-system-class-dhcp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-mgmt
      police cir 12000 kbps bc 4800000 bytes
    class copp-system-class-lacp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-lldp
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-udld
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-isis
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-msdp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-cdp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-fip
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bgp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-eigrp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-exception
      police cir 64 kbps bc 4800000 bytes
    class copp-system-class-glean
      police cir 4000 kbps bc 4800000 bytes
    class copp-system-class-hsrp-vrrp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-icmp-echo
      police cir 4000 kbps bc 3600000 bytes
    class copp-system-class-ospf
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-pim-register
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-rip
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-l3dest-miss
      police cir 64 kbps bc 3200000 bytes
    class copp-system-class-mcast-miss
```

```
    police cir 4000 kbps bc 3200000 bytes
  class copp-system-class-excp-ip-frag
    police cir 64 kbps bc 3200000 bytes
  class copp-system-class-excp-same-if
    police cir 64 kbps bc 3200000 bytes
  class copp-system-class-excp-ttl
    police cir 64 kbps bc 3200000 bytes
  class copp-system-class-default
    police cir 512 kbps bc 6400000 bytes
```

Customizable CoPP Policy

The copp-system-policy-customized policy is configured identically to the default policy, but can be customized for different class map information rates and burst sizes.

You cannot add or delete any of the class maps configured in this policy.

Important This policy is meant for advanced users. We recommend that you use extreme caution when configuring this policy and test it extensively before deploying it in your production network.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy-customized
    class copp-system-class-igmp
      police cir 1024 kbps bc 65535 bytes
    class copp-system-class-pim-hello
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bridging
      police cir 20000 kbps bc 4800000 bytes
    class copp-system-class-arp
      police cir 1024 kbps bc 3600000 bytes
    class copp-system-class-dhcp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-mgmt
      police cir 12000 kbps bc 4800000 bytes
    class copp-system-class-lacp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-lldp
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-udld
      police cir 2048 kbps bc 4800000 bytes
    class copp-system-class-isis
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-msdp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-cdp
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-fip
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-bgp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-eigrp
      police cir 9600 kbps bc 4800000 bytes
    class copp-system-class-exception
      police cir 64 kbps bc 4800000 bytes
    class copp-system-class-glean
      police cir 1024 kbps bc 4800000 bytes
    class copp-system-class-hsrp-vrrp
```

```
  police cir 1024 kbps bc 4800000 bytes
class copp-system-class-icmp-echo
  police cir 64 kbps bc 3600000 bytes
class copp-system-class-ospf
  police cir 9600 kbps bc 4800000 bytes
class copp-system-class-pim-register
  police cir 9600 kbps bc 4800000 bytes
class copp-system-class-rip
  police cir 9600 kbps bc 4800000 bytes
class copp-system-class-l3dest-miss
  police cir 64 kbps bc 3200000 bytes
class copp-system-class-mcast-miss
  police cir 256 kbps bc 3200000 bytes
class copp-system-class-excp-ip-frag
  police cir 64 kbps bc 3200000 bytes
class copp-system-class-excp-same-if
  police cir 64 kbps bc 3200000 bytes
class copp-system-class-excp-ttl
  police cir 64 kbps bc 3200000 bytes
class copp-system-class-default
  police cir 512 kbps bc 6400000 bytes
```

CoPP and the Management Interface

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

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

Licensing Requirements for CoPP

This feature does not require a license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for CoPP

- CoPP is a feature that is enabled by default in the switch. You cannot enable or disable CoPP.
- Only one control-plane policy can be applied at a time.
- Removing a CoPP policy applies the default CoPP policy. In this way, a CoPP policy is always applied.
- You cannot add or delete any classes or policies.
- You cannot change the order of the classes or remove a class from any policy.
- You cannot modify the default, the Scaled Layer-2, or the Scaled Layer 3 policies. However, you can modify the information rate and burst size of the classes in the customized policy.
- The customized policy configuration is the same as the default policy configuration, unless the customized policy has been modified.

- When upgrading from a previous release, the default CoPP policy is enabled by default on the switch.
- After modifying the customized policy or changing the applied policy, the statistical counters are reset.
- After you perform an ISSU, the statistical counters are reset.
- Cisco recommends that you use the default CoPP policy initially and then later determine which of the CoPP policies to use 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.
- Cisco recommends that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to use a different CoPP policy or modify the customized CoPP policy.
- All the traffic that you do not specify in the other class maps is put into the last class, the default class.
- 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).
- When a packet meets multiple exception conditions, CoPP matches the packet based on the order in which the CoPP ACLs are configured and match it only against a single class. This is an expected CoPP behavior.
- The copp-system-class-exception matches all IP data packets that are treated as exception packets (except TTL exception, IP Fragment exception and Same Interface exception packets) since the hardware itself is not capable of processing them. Control packets with ip-options are supposed to be caught by the respective protocol class and the protocol stack running on the switch is usually able to process them.

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 21: Default CoPP Parameters Settings

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

Configuring CoPP

Applying a CoPP Policy to the Switch

You can apply one of the following CoPP policies to the switch:

- Default CoPP Policy (copp-system-policy-default).
- Scaled Layer 2 CoPP Policy (copp-system-policy-scaled-l2).
- Scaled Layer 3 CoPP Policy (copp-system-policy-scaled-l3).
- Customized CoPP Policy (copp-system-policy-customized).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **control-plane**	Enters control-plane mode.
Step 3	switch(config-cp) # **service-policy input** *policy-map-name*	Applies the specified CoPP policy map. The *policy-map-name* can be copp-system-policy-default, copp-system-policy-scaled-l2, copp-system-policy-scaled-l3, or copp-system-policy-customized.
Step 4	switch(config-cp) # **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 apply a CoPP policy to the device:

```
switch# configure terminal
switch(config)# control-plane
switch(config-cp) # service-policy input copp-system-policy-default
switch(config-cp) # copy running-config startup-config
```

Modifying the Customized CoPP Policy

You can only modify the information rates and burst sizes of the class maps configured in this policy.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **policy-map type control-plane copp-system-policy-customized**	Enters configuration mode for the customized CoPP policy.
Step 3	switch(config-pmap)# **class** *class-map-name*	Specifies one of the 28 predefined class-maps listed in any CoPP predefined policy.
Step 4	switch(config-pmap-c)# **police cir** *rate-value* **kbps bc** *buffer-size* **bytes**	Configures the committed information rate (CIR) and committed burst size (BC). The range for cir is from 1 to 20480. The range for bc is from 1500 to 6400000.
Step 5	switch(config-pmap-c) # **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 modify the customized CoPP policy:

```
switch(config)# policy-map type control-plane copp-system-policy-customized
switch(config-pmap)# class copp-system-class-bridging
switch(config-pmap-c)# police cir 10000 kbps bc 2400000 bytes
```

Verifying the CoPP Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps.
show policy-map interface control-plane	Displays the policy values with associated class maps and drops per policy or class map.
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.

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)
      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-lisp
      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-rise
      match access-group name copp-system-p-acl-eigrp
      match access-group name copp-system-p-acl-lisp6
      match access-group name copp-system-p-acl-ospf6
      match access-group name copp-system-p-acl-rise6
```

```
    match access-group name copp-system-p-acl-eigrp6
    match access-group name copp-system-p-acl-otv-as
    match access-group name copp-system-p-acl-mac-l2pt
    match access-group name copp-system-p-acl-mpls-ldp
    match access-group name copp-system-p-acl-mpls-rsvp
    match access-group name copp-system-p-acl-mac-l3-isis
    match access-group name copp-system-p-acl-mac-otv-isis
    match access-group name copp-system-p-acl-mac-fabricpath-isis
    match protocol mpls router-alert
    set cos 7
    police cir 36000 kbps bc 250 ms
      conform action: transmit
      violate action: drop
    module 12:
      conformed 0 bytes,
        5-min offered rate 0 bytes/sec
        peak rate 0 bytes/sec
      violated 0 bytes,
        5-min violate rate 0 bytes/sec
        peak rate 0 bytes/sec
    module 14:
      conformed 0 bytes,
        5-min offered rate 0 bytes/sec
        peak rate 0 bytes/sec
      violated 0 bytes,
        5-min violate rate 0 bytes/sec
        peak rate 0 bytes/sec

  class-map copp-system-p-class-important (match-any)
    match access-group name copp-system-p-acl-cts
    match access-group name copp-system-p-acl-glbp
    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-wccp
    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-opflex
    match access-group name copp-system-p-acl-mac-lldp
    match access-group name copp-system-p-acl-mac-mvrp
    match access-group name copp-system-p-acl-mac-flow-control
    set cos 6
    police cir 1400 kbps bc 1500 ms
      conform action: transmit
      violate action: drop
    module 12:
      conformed 0 bytes,
        5-min offered rate 0 bytes/sec
        peak rate 0 bytes/sec
      violated 0 bytes,
        5-min violate rate 0 bytes/sec
        peak rate 0 bytes/sec
    module 14:
      conformed 0 bytes,
        5-min offered rate 0 bytes/sec
        peak rate 0 bytes/sec
      violated 0 bytes,
        5-min violate rate 0 bytes/sec
        peak rate 0 bytes/sec
....
```

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
```

Additional References for CoPP

This section provides additional information related to implementing CoPP.

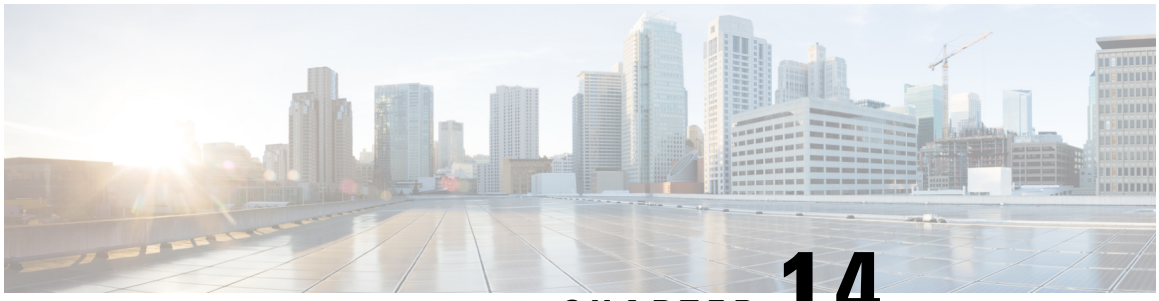
Related Documents

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	

Feature History for CoPP

Table 22: Feature History for CoPP

Feature Name	Feature Information
CoPP	Introduced in 5.1(3)N1(1)
CoPP	Additional IPv6 support in 5.2(1)N1(1)



CHAPTER 14

Configuring TCAM Carving

This chapter contains the following sections:

Information About TCAM Carving

The Ternary Content-Addressable Memory (TCAM) carving feature uses a template-based approach that enables you to modify the default region sizes of the TCAM. When the switch boots up, you see this default template, unless you have configured any other template. This table lists the types and sizes of various regions in a template.

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create
- Modify
- Delete
- Commit

Each template can be in one of the following states:

- Saved
- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 7.1(4)N1(1) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

- The committed template is saved in the startup configuration.

- The switch is rebooted.
- The committed template is used by the software.
- The template goes to the running state.

Note Prior to Cisco NX-OS Release 7.1(4)N1(1), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.
2. Any configuration after TCAM carving CLI is not applied.
3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2016
    Reason: Reload due to change in TCAM service-template
    Service:
    Version: 7.1(4)N1(1)

2) At 314447 usecs after Sun Jan 20 21:52:58 2016
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 7.1(4)N1(1)

3) At 20142 usecs after Sun Jan 20 21:27:33 2016
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 7.1(4)N1(1)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.
2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region

in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template. If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Creating a User-Defined Template

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource template** *template-name*	Creates a new template with the default region sizes. A maximum of 16 templates (plus the default) can be created. The *template-name* argument can be a maximum of 64 characters.

Example

This example shows how to create a user-defined template named qos-template:

```
switch# configure terminal
switch(config)# hardware profile tcam resource template qos-template
```

Modifying a User Defined Template

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource template** *template-name*	Creates a new template with the default region sizes. A maximum of 16 templates (plus the default) can be created. Use this command to enter template mode.

Example

This example shows how to modify a user-defined qos template.

```
switch# configure terminal
switch(config)# hardware profile tcam resource template qos-template
switch(config-tmpl) qos 64
```

Committing a User-Defined Template

You can commit a user-defined template.

Procedure

	Command or Action	Purpose
Step 1	Required: switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource service-template** *template-name*	Commits a previously defined template in the running image. After you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. If you agree to continue, the specified template is applied after the reboot. Otherwise, no changes are made to the TCAM regions and no template is committed.
Step 3	(Optional) switch# **show hardware profile tcam resource template**	Displays all templates. **Note** After the switch reloads, use this command to display the committed template.

Example

This example shows how to commit a user-defined template:

```
switch# configure terminal
switch(config)# hardware profile tcam resource service-template temp1

Details of the temp1 template you are trying to commit are as follows:

--------------------------------------------------------------------------------
Template name: temp1
Current state: Created

Region  Features  Size-allocated    Current-size  Current-usage  Available/free
--------------------------------------------------------------------------------
Vacl    Vacl               1024            1024             15            1009
Ifacl   Ifacl              1152            1152            209             943
Rbacl   Rbacl              1152            1152              3            1149
Qos     Qos                 448             448             30             418
Span    Span                 64              64              2              62
Sup     Sup                 256             256             58             198
--------------------------------------------------------------------------------


To finish committing the template, the system will do the following:
    1> Save running config :  "copy running-config startup-config"
    2> Reboot the switch   :  "reload"


--------------------------------------------------------------------------------
Do you really want to continue with RELOAD ? (y/n) [no] yes
System is still initializing
Configuration mode is blocked until system is ready
switch(config)# [16152.925385] Shutdown Ports..
[16152.959744]  writing reset reason 9
[snip]


/AFTER SWITCH RELOADS/

switch# show hardware profile tcam resource template
    Template   Type     State   Vacl  Ifacl   Rbacl  Qos  Span   Sup      TOTAL
-------------------------------------------------------------------------------
     default system   Created   1024   1152  1152    448    64   256      4096
       temp1   user Committed   1024   1152  1152    448    64   256      4096
       temp2   user   Created   1024   1152  1152    448    64   256      4096
-------------------------------------------------------------------------------
```

Deleting a Template

After creating a template, the template can be deleted. Deleting removes all the information about the template from the software.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no hardware profile tcam resource template** *template-name*	Deletes a user-defined template. Only saved templates can be deleted. Templates that are committed/running cannot be deleted. A template that is in the running configuration (same as the startup configuration) cannot be deleted. Any other user-defined template that is in a saved state can be deleted. The default template cannot be deleted.

Example

This example shows how to delete a template:

```
switch# configure terminal
switch(config)# no hardware profile tcam resource template qos-template
```

Verifying the TCAM Carving Configuration

To display TCAM carving configuration information, enter one of the following commands:

Command	Purpose
show hardware profile tcam resource template	Displays all templates.
show hardware profile tcam resource template name *template-name*	Displays a user-defined template.
show hardware profile tcam resource template default	Displays a default template.

INDEX

A

C

D

E

F

G

I

L

M

N

O

P

R

S

T

U

V