# Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 10.5(x)

**First Published:** 2024-07-26

# CONTENTS

**CHAPTER 26**

**CHAPTER 27**

# Preface

This preface includes the following sections:

- Audience, on page xxix
- Document Conventions, on page xxix
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xxx
- Documentation Feedback, on page xxx
- Communications, Services, and Additional Information, on page xxx

## Audience

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

## Document Conventions

Command descriptions use the following conventions:

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

| Convention | Description |
|---|---|
| variable | Indicates a variable for which you supply values, in context where italics cannot be used. |
| string | A nonquoted set of characters. Do not use quotation marks around the string or the string includes 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 that you must enter is in boldface screen font. |
| *italic screen font* | Arguments for which you supply values are in italic screen font. |
| < > | Nonprinting characters, such as passwords, are in angle brackets. |
| [ ] | Default responses to system prompts are in square brackets. |
| !, # | An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line. |

# Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

# Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

# Communications, Services, and Additional Information

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

- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.

- To submit a service request, visit Cisco Support.

- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.

- 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 gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

# Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

**CHAPTER 1**

# New and Changed Information

This chapter includes the new and changed features for the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 10.5(x).

- New and Changed Information, on page 1

# New and Changed Information

*Table 1: New and Changed Features*

| Feature | Description | Changed in Release | Where Documented |
|---|---|---|---|
| DACL extended support | Extended support for the DACL feature on Cisco Nexus 9300-FX3, GX, GX2, H2R, and H1 Series switches | 10.5(2)F | Guidelines and Limitations for Per-User DACL Support for 802.1X, on page 254 |
| uRPF support | Added uRPF support on Cisco Nexus 9800 Series switches and N9K-C9336C-K1 switches. | 10.5(2)F | Guidelines and Limitations for Unicast RPF, on page 548 |
| Extended RSA key size support | Extended support for RSA key sizes to 4096 bits for SSH and to 3072 and 4096 bits for cryptographic certificates. | 10.5(2)F | Generating SSH Server Keys, on page 141<br><br>Configuring SSH Passwordless File Copy, on page 146<br><br>Generating an RSA Key Pair, on page 175 |

| Feature | Description | Changed in Release | Where Documented |
|---|---|---|---|
| Dot1x for Voice VLAN and critical authentication | Added support for the 802.1X Voice VLAN feature to enable multi-domain 802.1X authentication on a single port, providing authentication support for both VoIP phone and data client behind it. | 10.5(2)F | About 802.1X for Voice VLAN, on page 247<br><br>Critical Authentication, on page 247<br><br>802.1X Guidelines and Limitations for Voice VLAN, on page 253<br><br>Configuring 802.1X for Voice VLAN, on page 259 |
| Security Group ACL (SGACL) Feature interaction support | Added support of Security Group ACL for the ESI, VXLAN-TE, VXLAN-PBR, CloudSec (DCI), and TRM features. | 10.5(1)F | Guidelines and Limitations for IP ACLs, on page 307 |

# 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.

This chapter includes the following sections:

# Licensing Requirements

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

# Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

# 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.

For more information, see the Configuring AAA, on page 19 chapter.

# 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.

For more information, see the Configuring TACACS+, on page 83 chapter and the Configuring RADIUS, on page 53 chapter.

# LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP allows a single access control server (the LDAP daemon) to provide authentication and authorization independently.

For more information, see the Configuring LDAP, on page 115 chapter.

# 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.

For more information, see the Configuring SSH and Telnet, on page 137 chapter.

# User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the Cisco NX-OS device. Role-based access control (RBAC) allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

For more information, see the Configuring User Accounts and RBAC, on page 219 chapter.

# 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.

For more information, see the chapter.

# MAC ACLs

MAC ACLs are ACLs that filter traffic using the information in the Layer 2 header of each packet. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that a MAC ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

# VACLs

A VLAN ACL (VACL) is one application of an IP ACL or MAC ACL. You can configure VACLs to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

For more information, see the chapter.

# 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.

Dynamic ARP inspection (DAI) and IP Source Guard (IPSG) also use information stored in the DHCP snooping binding database.

# Dynamic ARP Inspection

Dynamic ARP inspection (DAI) ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco NX-OS device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports.

- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination.

• Drops invalid ARP packets.

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a DHCP snooping binding database. This database can also contain static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

# IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches one of two sources of IP and MAC address bindings:

• Entries in the DHCP snooping binding table.

• Static IP source entries that you configure.

Filtering on trusted IP and MAC address bindings helps prevent attacks that rely on spoofing the IP address of a valid host. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

# Password Encryption

The Advanced Encryption Standard (AES) password encryption feature stores all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) in the strong and reversible type-6 encrypted format. A primary encryption key is used to encrypt and decrypt the passwords. You can also use this feature to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

For more information, see the Configuring Password Encryption, on page 515 chapter.

# Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication.

For more information, see the Configuring Keychain Management, on page 523 chapter.

# Control Plane Policing

The Cisco NX-OS device provides control plane policing to prevent denial-of-service (DoS) attacks from impacting performance. The supervisor module of the Cisco NX-OS device has both the management plane and control plane and is critical to the operation of the network. Any disruption to the supervisor module would result in serious network outages. Excessive traffic to the supervisor module could overload it and slow down the performance of the entire Cisco NX-OS device. Attacks on the supervisor module can be of various types such as, denial-of-service (DoS) attacks that generate IP traffic streams to the control plane at a very

high rate. These attacks result in the control plane spending a large amount of time in handling these packets, which makes the control plane unable to process genuine traffic.

For more information, see the Configuring Control Plane Policing, on page 561 chapter.

# Rate Limits

Rate limits can prevent redirected packets for egress exceptions from overwhelming the supervisor module on a Cisco NX-OS device.

For more information, see the Configuring Rate Limits, on page 601 chapter.

# Software Image

The Cisco NX-OS software consists of one NXOS software image.

# Virtual Device Contexts

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

# SGT Tagged Packet

Beginning with Cisco NX-OS Release 10.2(2)F, the Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches can be used as a L2 or L3 transit device to forward a SGT tagged packet. Cisco NX-OS N9K can pass the packet without stripping or altering the SGT.

Beginning with Cisco NX-OS Release 10.3(3)F, SGT forwarding is supported on Cisco Nexus 9500 switches with N9K-X97160YC-EX line card.

**Note**    NX-OS N9K does not do any SGT enforcement. It also does not generate or rewrite SGT/DGT info in the packets. Packets are received and transmitted out with the same tags.

# TLS Protocol Support

The Transport Layer Security (TLS) protocol is a cryptographic protocol designed to provide secure communication over a computer network.

Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus application supports Transport Layer Security (TLS) version 1.3 by default.

The following applications supports TLSv1.3 for faster and secure communication:

  • NX-API: For more details, see NX-API Management Commands.

- gNMI: For more details, see gNMI - Management Interface.

- Secure syslog: For more details, see Configuring System Message Logging.

- RadSec: For more details, see Guidelines and Limitations for RadSec, on page 58.

- Copy utilities (HTTPS option): For more details, see Using the Device File Systems, Directories, and Files.

- Callhome: For more details, see Configuring Smart Call Home.

- Smart Licensing: For more details, see Smart Licensing Using Policy.

- 802.1x: For more details, see 802.1X Guidelines and Limitations, on page 250.

- gRPC agent: For more details, see gRPC Agent.

- LDAP: For more details, see Guidelines and Limitations for LDAP, on page 118.

**CHAPTER 3**

# Configuring FIPS

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

This chapter includes the following sections:

## About FIPS

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

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

## FIPS Self-Tests

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

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

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

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

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

Conditional self-tests include the following:

**Pair-wise consistency test**

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

**Continuous random number generator test**

This test is run when a random number is generated.

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

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

# FIPS Error State

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

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

# Prerequisites for FIPS

FIPS has the following prerequisites:

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

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

- Delete all SSH server RSA1 key-pairs.

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

# Guidelines and Limitations for FIPS

FIPS has the following configuration guidelines and limitations:

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

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

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

- When RadSec is enabled, Radius doesn't need to be disabled when FIPS is enabled.

- When the **fips mode enable** command is executed after an ASCII reload, you need to reload the Cisco NX-OS switch after executing the **copy running-config startup-config** command.

# Default Settings for FIPS

This table lists the default settings for FIPS parameters.

**Table 2: Default FIPS Parameters**

| Parameters | Default |
|------------|---------|
| FIPS mode | Disabled |

# Configuring FIPS

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

# Enabling FIPs Mode

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

**Before you begin**

Ensure that you are in the default VDC.

**Procedure**

|  | Command or Action | Purpose |
|--------|-------------------|---------|
| Step 1 | **configure terminal** <br><br> **Example:** <br> ```switch# configure terminal``` <br> ```switch(config)#``` | Enters global configuration mode. |

| | | Command or Action | Purpose |
|---|---|---|---|
| Step 2 | | **fips mode enable**<br><br>**Example:**<br>`switch(config)# fips mode enable` | Enables FIPS mode.<br><br>**Note** **fips mode enable** can be entered only when all LCs are online or else it leads to LC failure. |
| Step 3 | | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| Step 4 | | (Optional) **show fips status**<br><br>**Example:**<br>`switch# show fips status`<br>`FIPS mode is enabled` | Displays the status of FIPS mode. |
| Step 5 | | Required: **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |
| Step 6 | | Required: **reload**<br><br>**Example:**<br>`switch# reload` | Reloads the Cisco NX-OS device.<br><br>**Note** After you enable FIPS, a reboot is required for the system to operate in FIPS mode. |

# Disabling FIPS

You can disable FIPS mode on the device.

**Before you begin**

Ensure that you are in the default VDC.

**Procedure**

| | | Command or Action | Purpose |
|---|---|---|---|
| Step 1 | | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | | **no fips mode enable**<br><br>**Example:**<br>`switch(config)# no fips mode enable` | Disables FIPS mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional)  **show fips status**<br><br>**Example:**<br><br>`switch# show fips status`<br>`FIPS mode is disabled` | Displays the status of FIPS mode. |
| **Step 5** | **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |
| **Step 6** | **reload**<br><br>**Example:**<br><br>`switch# reload` | Reloads the Cisco NX-OS device. |

# Verifying the FIPS Configuration

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

| Command | Purpose |
|---|---|
| **show fips status** | Displays the status of the FIPS feature. |

For detailed information about the fields in the output from this command, see the *Cisco Nexus 9000 Series NX-OS Security Command Reference*.

# Create 2048 bit RSA Key

Steps to create a 2048 bit RSA key:

- `N9k-Switch# conf t`

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

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

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

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

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

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

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

```
**************************************
rsa Keys generated:Wed Apr 28 13:05:18 2021
ssh-rsa  AAAAB3NzaC1yc2EAAAADAQABAAABAQDHpxEgZ9LwmbOEpJeJtLwqedmTLkZV7Setxb9D4xgO
p2o2f6wt/48bPp/vLDGsxTF2PtLRtRSSDFNSQmkw9bg+MXvTpgNivdxWLjxtwo3YpYwPkBiReVmyrFgE
UuBmV/sDfhJpHXLoH9lR2+y0L5w1OG3cJxMe30TI37O3M8fZPjrAtHgkUubfEpiTbcyEw+aIHf+chyoR
eDJxcEdnlboiTDFR0/+jMUUM/vMtxd5x5DH3AO7htA/i8lvskrReR1CpX1sOOdcshmS57EEuEzR9cs+w
KSftQh6vLD802207T6+J7/+cXMVNQEbq0mCSzeTmOsuIQe8u9ZC24pgYzZ19
bitcount:2048
fingerprint:
SHA256:Am9861AIq5MzfSPQr4ZXGe0f5M9crnhk7HVZBXhMVBo
**************************************
could not retrieve dsa key information
**************************************
could not retrieve ecdsa key information
**************************************
```

# Configuration Example for FIPS

The following example shows how to enable FIPS mode:

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

# Additional References for FIPS

This section includes additional information related to implementing FIPS.

**Related Documents**

| Related Topic | Document Title |
| --- | --- |
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |
| Command reference | *Cisco Nexus 9000 Series NX-OS Security Command Reference* |

**Standards**

| Standards | Title |
|-----------|-------|
| FIPS 140-2 | Security Requirements for Cryptographic Modules |

# Configuring AAA

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

This chapter includes the following sections:

# About AAA

This section includes information about AAA on Cisco NX-OS devices.

## AAA Security Services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing a Cisco NX-OS device. Cisco NX-OS devices support Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, Cisco NX-OS devices 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 Cisco NX-OS device 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, depending on the security protocol that you select, encryption.

Authentication is the process of verifying the identity of the person or device accessing the Cisco NX-OS device, which is based on the user ID and password combination provided by the entity trying to access the Cisco NX-OS device. Cisco NX-OS devices allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

**Authorization**

Provides access control.AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the Cisco NX-OS software 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.

The accounting feature tracks and maintains a log of every management session used to access the Cisco NX-OS device. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

**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:

- It is easier to manage user password lists for each Cisco NX-OS device in the fabric.

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

- You can centrally manage the accounting log for all Cisco NX-OS devices in the fabric.

- It is easier to manage user attributes for each Cisco NX-OS device in the fabric than using the local databases on the Cisco NX-OS devices.

# 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 implements the same AAA protocol. The purpose of a server group is to provide for failover servers in case 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, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Cisco NX-OS device encounters errors from the servers in the first group, it tries the servers in the next server group.

# AAA Service Configuration Options

The AAA configuration in Cisco NX-OS devices is service based, which means that 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

This table provides the related CLI command for each AAA service configuration option.

**Table 3: AAA Service Configuration Commands**

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

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

**All RADIUS servers**

Uses the global pool of RADIUS servers for authentication.

**Specified server groups**

Uses specified RADIUS, TACACS+, or LDAP server groups you have configured for authentication.

**Local**

Uses the local username or password database for authentication.

**None**

Specifies that no AAA authentication be used.

**Note**    If you specify the all RADIUS servers method, rather than a specified server group method, the Cisco NX-OS device chooses 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 NX-OS device.

This table shows the AAA authentication methods that you can configure for the AAA services.

*Table 4: AAA Authentication Methods for AAA Services*

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

**Note**   For console login authentication, user login authentication, and user management session accounting, the Cisco NX-OS device tries each option in the order specified. The local option is the default method when other configured options fail. You can disable the local option for the console or default login by using the **no aaa authentication login** {**console** | **default**} **fallback error local** command.

# Authentication and Authorization Process for User Login

*Figure 1: Authorization and Authentication Flow for User Login*

This figure shows a flow chart of the authentication and authorization process for user login.
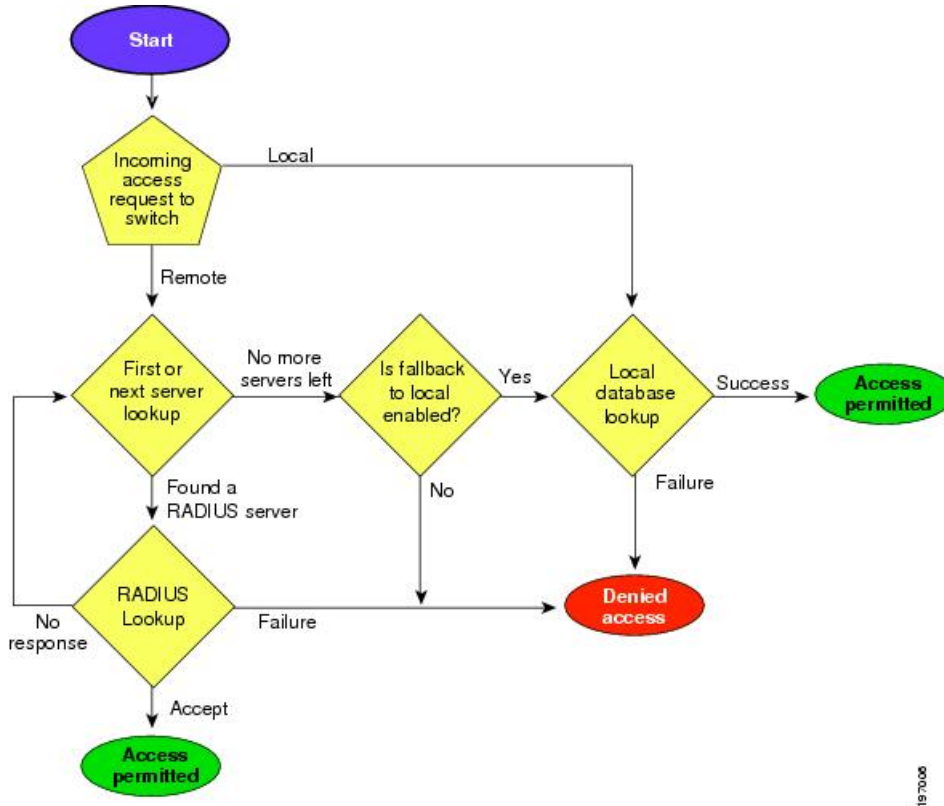


The following list explains the process:

- When you log in to the required Cisco NX-OS device, you can use the Telnet, SSH, or console login options.

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

  - If the AAA server fails to respond, 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, unless fallback to local is disabled for the console login.

- If the Cisco NX-OS device successfully authenticates you through a remote AAA server, then the following possibilities apply:

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

  - If the AAA server protocol is TACACS+, then 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 NX-OS device logs you in and assigns you the roles configured in the local database.

**Note** "No more server groups left" means that there is no response from any server in all server groups. "No more servers left" means that there is no response from any server within this server group.

# AES Password Encryption and Primary Encryption Keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. To start using type-6 encryption, you must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in type-6 encrypted format, unless you disable type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

# Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that at least one RADIUS, TACACS+, or LDAP server is reachable through IP.

- Ensure that the Cisco NX-OS device is configured as a client of the AAA servers.

- Ensure that the secret key is configured on the Cisco NX-OS device and the remote AAA servers.

- Ensure that the remote server responds to AAA requests from the Cisco NX-OS device.

# Guidelines and Limitations for AAA

AAA has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.2(1)F, SNMPV3 attributes can be mentioned before the shell:roles attribute in cisco-av-pair.

- LDAP does not support 'snmpv3' attributes.

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

- Cisco Nexus 9000 Series switches support the **aaa authentication login ascii-authentication** command only for TACACS+ (and not for RADIUS).

- If you modify the default login authentication method (without using the **local** keyword), the configuration overrides the console login authentication method. To explicitly configure the console authentication method, use the **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} command.

- The **login block-for** and **login quiet-mode** configuration mode commands are renamed to **system login block-for** and **system login quiet-mode**, respectively.

- When you use the **system login quiet-mode access-class QUIET_LIST** command, you must ensure that the access list is correctly defined to only block the specified traffic. For example, if you need to block only the user logins from untrusted hosts, then the access list should specify ports 22, 23, 80, and 443 corresponding to SSH, telnet, and HTTP-based access from those hosts.

- Beginning with Cisco NX-OS Release 10.2(2)F, a new desynchronization CLI is introduced to provide you an option to disable the user synchronization between the SNMP and the security components. For more information, refer to the *Configuring SNMP* chapter in the *System Management Configuration Guide*.

  For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- When the desynchronization CLI is enabled, remote users will not be synced to SNMP database.

- The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a) will not have a corresponding SNMPv3 profile when the desynchronization CLI is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization CLI along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

- Beginning with Cisco NX-OS Release 10.3(1)F, AAA is supported on the Cisco Nexus 9808 switches.

  - Beginning with Cisco NX-OS Release 10.4(1)F, AAA is supported on Cisco Nexus X98900CD-A, and X9836DM-A line cards with 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, AAA is supported on the Cisco Nexus 9804 switches, X98900CD-A, and X9836DM-A line cards.

• Beginning with Cisco NX-OS release 10.4(3)F, support for SSH based authorization of X.509 certificates using TACACS+ server is being provided on the Cisco Nexus 9000 Series platform switches. This feature can be enabled using **aaa authorization ssh-certificate default group** *tac-group-name* command. For more information, see Configuring AAA SSH-Cert-Authorization on TACACS Servers, on page 39.

# Default Settings for AAA

This table lists the default settings for AAA parameters.

*Table 5: Default AAA Parameter Settings*

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

# Configuring AAA

This section describes the tasks for configuring AAA on Cisco NX-OS devices.

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

**Note**   Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

# Process for Configuring AAA

Follow these steps to configure AAA authentication and accounting:

1. If you want to use remote RADIUS, TACACS+, or LDAP servers for authentication, configure the hosts on your Cisco NX-OS device.

2. Configure console login authentication methods.

3. Configure default login authentication methods for user logins.

4. Configure default AAA accounting default methods.

# Configuring Console Login Authentication Methods

This section describes how to configure the authentication methods for the console login.

The authentication methods include the following:

- Global pool of RADIUS servers

- Named subset of RADIUS, TACACS+, or LDAP servers

- Local database on the Cisco NX-OS device

- Username only (none)

The default method is local, but you have the option to disable it.

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

**Note** If you perform a password recovery when remote authentication is enabled, local authentication becomes enabled for console login as soon as the password recovery is done. As a result, you can log into the Cisco NX-OS device through the console port using the new password. After login, you can continue to use local authentication, or you can enable remote authentication after resetting the admin password configured at the AAA servers. For more information about the password recovery process, see the *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.*

**Before you begin**

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **aaa authentication login console** {**group** *group-list* [**none**] \| **local** \| **none**} | Configures login authentication methods for the console. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>switch(config)# **aaa authentication login console group radius** | The *group-list* argument consists of a space-delimited list of group names. The group names are the following:<br><br>**radius**<br>    Uses the global pool of RADIUS servers for authentication.<br>*named-group*<br>    Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.<br><br>The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used.<br><br>The default console login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login. |
| **Step 3** | **exit**<br>**Example:**<br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show aaa authentication**<br>**Example:**<br>switch# **show aaa authentication** | Displays the configuration of the console login authentication methods. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

# Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers

- Named subset of RADIUS, TACACS+, or LDAP servers

- Local database on the Cisco NX-OS device

- Username only

The default method is local, but you have the option to disable it.

**Before you begin**

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **aaa authentication login default** {**group** *group-list* [**none**] \| **local** \| **none**}<br><br>**Example:**<br>`switch(config)# aaa authentication login default group radius` | Configures the default authentication methods.<br><br>The *group-list* argument consists of a space-delimited list of group names. The group names are the following:<br><br>   • **radius**—Uses the global pool of RADIUS servers for authentication.<br><br>   • *named-group*—Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.<br><br>The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used. The default login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login.<br><br>You can configure one of the following:<br><br>   • AAA authentication groups<br><br>   • AAA authentication groups with no authentication<br><br>   • Local authentication<br><br>   • No authentication |

| | Command or Action | Purpose |
|---|---|---|
| | **Note** The **local** keyword is not supported (and is not required) when configuring AAA authentication groups because local authentication is the default if remote servers are unreachable. For example, if you configure **aaa authentication login default group g1**, local authentication is tried if you are unable to authenticate using AAA group g1. In contrast, if you configure **aaa authentication login default group g1 none**, no authentication is performed if you are unable to authenticate using AAA group g1.<br><br>**Note** Cisco NX-OS AAA authentication does not support hashed key and only supports Type 6/7 keys. | |
| Step 3 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| Step 4 | (Optional) **show aaa authentication**<br><br>**Example:**<br>`switch# show aaa authentication` | Displays the configuration of the default login authentication methods. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Disabling Fallback to Local Authentication

By default, if remote authentication is configured for console or default login and all AAA servers are unreachable (resulting in an authentication error), the Cisco NX-OS device falls back to local authentication to ensure that users aren't locked out of the device. However, you can disable fallback to local authentication in order to increase security.

⚠️

**Caution**  Disabling fallback to local authentication can lock your Cisco NX-OS device, forcing you to perform a password recovery in order to gain access. To prevent being locked out of the device, we recommend that you disable fallback to local authentication for only the default login or the console login, not both.

**Before you begin**

Configure remote authentication for the console or default login.

**Procedure**

|        | **Command or Action** | **Purpose** |
|--------|----------------------|-------------|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **no aaa authentication login** {**console** \| **default**} **fallback error local**<br><br>**Example:**<br><br>`switch(config)# `**`no aaa authentication`**<br>**`login console fallback error local`** | Disables fallback to local authentication for the console or default login if remote authentication is configured and all AAA servers are unreachable.<br><br>The following message appears when you disable fallback to local authentication:<br><br>`"WARNING!!! Disabling fallback can`<br>`lock your switch."` |
| **Step 3** | (Optional) **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show aaa authentication**<br><br>**Example:**<br><br>`switch# `**`show aaa authentication`** | Displays the configuration of the console and default login authentication methods. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>**`startup-config`** | Copies the running configuration to the startup configuration. |

# Enabling the Default User Role for AAA Authentication

You can allow remote users who do not have a user role to log in to the Cisco NX-OS device through a RADIUS or TACACS+ remote authentication server using a default user role. When you disable the AAA default user role feature, remote users who do not have a user role cannot log in to the device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal```<br>```switch(config)#``` | Enters configuration mode. |
| **Step 2** | **aaa user default-role**<br><br>**Example:**<br><br>```switch(config)# aaa user default-role``` | Enables the default user role for AAA authentication. The default is enabled.<br><br>You can disable the default user role feature by using the **no** form of this command. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>```switch(config)# exit```<br>```switch#``` | Exits configuration mode. |
| **Step 4** | (Optional) **show aaa user default-role**<br><br>**Example:**<br><br>```switch# show aaa user default-role``` | Displays the AAA default user role configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```switch# copy running-config```<br>```startup-config``` | Copies the running configuration to the startup configuration. |

# Enabling Login Authentication Failure Messages

When you log in, the login is processed by rolling over to the local user database if the remote AAA servers do not respond. In such cases, the following messages display on the user's terminal if you have enabled login failure messages:

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

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal```<br>```switch(config)#``` | Enters configuration mode. |
| **Step 2** | **aaa authentication login error-enable**<br><br>**Example:** | Enables login authentication failure messages. The default is disabled. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# **aaa authentication login error-enable** | |
| **Step 3** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show aaa authentication**<br><br>**Example:**<br><br>switch# **show aaa authentication** | Displays the login failure message configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# copy running-config<br>startup-config | Copies the running configuration to the startup configuration. |

# Logging Successful and Failed Login Attempts

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

**Procedure**

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

| | Command or Action | Purpose |
|---|---|---|
| | ```
switch(config)# login on-success log
switch(config)# logging level authpriv
6
switch(config)# logging level daemon 6
``` | following syslog message appears after the successful login:<br><br>AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00<br><br>**Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3. |
| Step 4 | (Optional) **show login on-failure log**<br><br>**Example:**<br><br>```
switch(config)# show login on-failure
log
``` | Displays whether the switch is configured to log failed authentication messages to the syslog server. |
| Step 5 | (Optional) **show login on-successful log**<br><br>**Example:**<br><br>```
switch(config)# show login on-successful
 log
``` | Displays whether the switch is configured to log successful authentication messages to the syslog server. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```
switch(config)# copy running-config
startup-config
``` | Copies the running configuration to the startup configuration. |

# Configuring Login Block Per User

Ensure that the switch is in global configuration mode.

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

**Note** From Release 9.3(7), you can configure login block for remote users.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>```
switch# configure terminal
``` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **aaa authentication rejected** *attempts***in***seconds***ban***seconds*<br><br>**Example:**<br>switch(config)# **aaa authentication rejected 3 in 20 ban 300** | Configures login parameters to block a user.<br><br>**Note**    Use **no aaa authentication rejected** command to revert to the default login parameters. |
| Step 3 | **exit**<br><br>**Example:**<br>switch(config)# **exit** | Exits to privileged EXEC mode. |
| Step 4 | (Optional) **show running config**<br><br>**Example:**<br>switch# **show running config** | Displays the login parameters. |
| Step 5 | **show aaa local user blocked**<br><br>**Example:**<br>switch# **show aaa local user blocked** | Displays the blocked local users. |
| Step 6 | **clear aaa local user blocked {username user\| all}**<br><br>**Example:**<br>switch(config)# **switch# clear aaa local user blocked username testuser** | Clears the blocked local users.<br><br>all –Clears all the blocked local users. |
| Step 7 | **show aaa user blocked**<br><br>**Example:**<br>switch(config)# **show aaa user blocked** | Displays all blocked local and remote users. |
| Step 8 | (Optional) **clear aaa user blocked{username user\| all}**<br><br>**Example:**<br>switch# **clear aaa user blocked username testuser** | Clears all blocked local and remote users.<br><br>all – Clears all the blocked local and remote users. |

**Example**

✎

**Note**    Only network-admin, and vdc-admin have privileges to run the show and clear commands.

The following example shows how to configure the login parameters to block a user for 300 seconds when three login attempts fail within a period of 20 seconds:

```
switch(config)# aaa authentication rejected 3 in 20 ban 300
switch# show run | i rejected
aaa authentication rejected 3 in 20 ban 300
```

```
switch# show aaa local user blocked
Local-user              State
testuser                Watched (till 11:34:42 IST Nov 12 2020)
switch# clear aaa local user blocked username testuser
switch# show aaa user blocked
Local-user              State
testuser                Watched (till 11:34:42 IST Nov 12 2020)
switch# clear aaa user blocked username testuser
```

# Enabling CHAP Authentication

The Cisco NX-OS software supports the Challenge Handshake Authentication Protocol (CHAP), a challenge-response authentication protocol that uses the industry-standard Message Digest (MD5) hashing scheme to encrypt responses. You can use CHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable CHAP, you need to configure your RADIUS or TACACS+ server to recognize the CHAP vendor-specific attributes (VSAs).

**Note**    Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors. For example:

```
2017 Jun 14 16:14:15 N9K-1 %RADIUS-2-RADIUS_NO_AUTHEN_INFO: ASCII authentication not supported
2017 Jun 14 16:14:16 N9K-1 %AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed from
192.168.12.34 - dcos_sshd[16804]
```

This table shows the RADIUS and TACACS+ VSAs required for CHAP.

*Table 6: CHAP RADIUS and TACACS+ VSAs*

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

**Before you begin**

Disable AAA ASCII authentication for logins.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal```<br>```switch(config)#``` | Enters configuration mode. |
| **Step 2** | **no aaa authentication login ascii-authentication**<br><br>**Example:**<br><br>```switch(config)# no aaa authentication```<br>```login ascii-authentication``` | Disables ASCII authentication. |
| **Step 3** | **aaa authentication login chap enable**<br><br>**Example:**<br><br>```switch(config)# aaa authentication login```<br>```chap enable``` | Enables CHAP authentication. The default is disabled.<br><br>**Note**  You cannot enable both CHAP and MSCHAP or MSCHAP V2 on your Cisco NX-OS device. |
| **Step 4** | (Optional) **exit**<br><br>**Example:**<br><br>```switch(config)# exit```<br>```switch#``` | Exits configuration mode. |
| **Step 5** | (Optional) **show aaa authentication login chap**<br><br>**Example:**<br><br>```switch# show aaa authentication login```<br>```chap``` | Displays the CHAP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```switch# copy running-config```<br>```startup-config``` | Copies the running configuration to the startup configuration. |

# Enabling MSCHAP or MSCHAP V2 Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. The Cisco NX-OS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a Cisco NX-OS device through remote authentication RADIUS servers. If you configure a TACACS+ group with MSCHAP V2, the AAA default login authentication uses the next configured method, or the local method, if no other server group is configured.

| **Note** | The Cisco NX-OS software may display the following message: |
|---|---|
| | " Warning: MSCHAP V2 is supported only with Radius." |
| | This warning message is informational only and does not affect MSCHAP V2 operation with RADIUS. |

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable MSCHAP or MSCHAP V2, you need to configure your RADIUS server to recognize the MSCHAP and MSCHAP V2 vendor-specific attributes (VSAs).

This table shows the RADIUS VSAs required for MSCHAP.

**Table 7: MSCHAP and MSCHAP V2 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 or MSCHAP V2 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 or MSCHAP V2 user in response to the challenge. It is only used in Access-Request packets. |

**Before you begin**

Disable AAA ASCII authentication for logins.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **no aaa authentication login ascii-authentication**<br><br>**Example:**<br><br>`switch(config)# no aaa authentication login ascii-authentication` | Disables ASCII authentication. |
| **Step 3** | **aaa authentication login {mschap | mschapv2} enable**<br><br>**Example:**<br><br>`switch(config)# aaa authentication login mschap enable` | Enables MSCHAP or MSCHAP V2 authentication. The default is disabled.<br><br>**Note**  You cannot enable both MSCHAP and MSCHAP V2 on your Cisco NX-OS device. |

|        | **Command or Action** | **Purpose** |
|--------|------------------------|-------------|
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show aaa authentication login** {**mschap** \| **mschapv2**}<br><br>**Example:**<br>`switch# show aaa authentication login mschap` | Displays the MSCHAP or MSCHAP V2 configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring AAA Authorization on LDAP Servers

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

**Before you begin**

Enable LDAP.

**Procedure**

|        | **Command or Action** | **Purpose** |
|--------|------------------------|-------------|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] \| **local** \| **none**}<br><br>**Example:**<br>`switch(config)# aaa authorization ssh-certificate default group ldap1 ldap2` | Configures the default AAA authorization method for the LDAP servers.<br><br>The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.<br><br>The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, |

| | Command or Action | Purpose |
|---|---|---|
| | | and the **none** method specifies that no AAA authorization be used. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits global configuration mode. |
| **Step 4** | (Optional)  **show aaa authorization** [**all**]<br><br>**Example:**<br><br>`switch# show aaa authorization` | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| **Step 5** | (Optional)  **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Enabling TACACS+ , on page 89

# Configuring AAA SSH-Cert-Authorization on TACACS Servers

To configure AAA SSH-Cert-Authorization on TACACS Servers, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}<br><br>**Example:**<br><br>`switch(config)# `**`aaa authorization`**<br>**`ssh-certificate default group tac1`** | Configures the default AAA authorization-method for SSH request having X509 certificate as TACACS server-group(s).<br><br>The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.<br><br>The *group-list* argument consists of a space-delimited list of TACACS server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, |

| | Command or Action | Purpose |
|---|---|---|
| | | and the **none** method specifies that no AAA authorization be used. |
| Step 3 | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits global configuration mode. |
| Step 4 | (Optional) **show aaa authorization** [**all**]<br><br>**Example:**<br><br>`switch# show aaa authorization` | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring AAA Accounting Default Methods

Cisco NX-OS software supports TACACS+ and RADIUS methods for accounting. Cisco NX-OS devices 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 NX-OS 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**

Configure RADIUS or TACACS+ server groups, as needed.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters configuration mode. |
| Step 2 | **aaa accounting default** {**group** *group-list* \| **local**}<br><br>**Example:**<br><br>switch(config)# **aaa accounting default group radius** | Configures the default accounting method.<br><br>The *group-list* argument consists of a space-delimited list of group names. The group names are the following:<br><br>• **radius**—Uses the global pool of RADIUS servers for accounting.<br><br>• *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting.<br><br>The **local** method uses the local database for accounting.<br><br>The default method is **local**, which is used when no server groups are configured or when all the configured server groups fail to respond. |
| Step 3 | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| Step 4 | (Optional) **show aaa accounting**<br><br>**Example:**<br><br>switch# **show aaa accounting** | Displays the configuration AAA accounting default methods. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

# Using AAA Server VSAs with Cisco NX-OS Devices

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

## About VSAs

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 separator value *
```

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

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

## VSA Format

The following VSA protocol options are supported by the Cisco NX-OS software:

**Shell**

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

**Accounting**

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

The following attributes are supported by the Cisco NX-OS software:

**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. For example, if you belong to role network-operator and network-admin, the value field would be network-operator network-admin. This subattribute is sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These examples use the roles attribute:

```
shell:roles=network-operator network-admin
```

```
shell:roles*network-operator network-admin
```

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

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

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

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

**accountinginfo**

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

## Specifying Cisco NX-OS 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 NX-OS 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.

The SNMPv3 attributes should come together, either before the shell attributes or after. You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
snmpv3:auth="SHA" priv="AES-128" shell:roles="network-admin" shell:priv-lvl=15
```

```
shell:roles="network-admin" shell:priv-lvl=15 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.

# Configuring Secure Login Features

## Configuring Login Parameters

You can configure login parameters to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected and slow down dictionary attacks by enforcing a quiet period if multiple failed connection attempts are detected.

**Note** This feature restarts if a system switchover occurs or the AAA process restarts.

**Note** The **login block-for** and **login quiet-mode** configuration mode commands have been renamed to **system login block-for** and **system login quiet-mode**, respectively.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br><br>**Example:** <br><br>switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | [**no**] **system login block-for** *seconds* **attempts** *tries* **within** *seconds* | Configures the quiet mode time period. The range for all arguments is from 1 to 65535. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`switch(config)# system login block-for 100 attempts 2 within 60` | The example shows how to configure the switch to enter a 100-second quiet period if 2 failed login attempts are exceeded within 60 seconds.<br><br>After you enter this command, all login attempts made through Telnet or SSH are denied during the quiet period. Access control lists (ACLs) are not exempt from the quiet period until the **system** command is entered.<br><br>**Note**    You must enter this command before any other login command can be used. |
| **Step 3** | (Optional) [**no**] **system login quiet-mode access-class** *acl-name*<br>**Example:**<br>`switch(config)# system login quiet-mode access-class myacl` | Specifies an ACL that is to be applied to the switch when it changes to quiet mode. When the switch is in quiet mode, all login requests are denied, and the only available connection is through the console. |
| **Step 4** | (Optional) **show system login** [**failures**]<br>**Example:**<br>`switch(config)# show system login` | Displays the login parameters. The **failures** option displays information related only to failed login attempts. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

## Restricting User Login Sessions

You can restrict the maximum number of simultaneous login sessions per user. Doing so prevents users from having multiple unwanted sessions and solves the potential security issue of unauthorized users accessing a valid SSH or Telnet session.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal` | Enters global configuration mode. |
| **Step 2** | [**no**] **user max-logins** *max-logins*<br>**Example:**<br>`switch(config)# user max-logins 1` | Restricts the maximum number of simultaneous login sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, only one Telnet or SSH session is allowed per user. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** The configured login limit applies to all users. You cannot set a different limit for individual users. |
| **Step 3** | (Optional) **show running-config all | i max-login**<br><br>**Example:**<br><br>switch(config)# **show running-config all | i max-login** | Displays the maximum number of login sessions allowed per user. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch(config)# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

## Restricting the Password Length

You can restrict the minimum and maximum length of the user password. This feature enables you to increase system security by forcing the user to provide a strong password.

### Before you begin

You must enable password strength checking using the **password strength-check** command. If you restrict the password length but do not enable password strength checking and the user enters a password that is not within the restricted length, an error appears, but a user account is created. To enforce the password length and prevent a user account from being created, you must enable password strength checking and restrict the password length.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | [**no**] **userpassphrase** {**min-length** *min-length* \| **max-length** *max-length*}<br><br>**Example:**<br><br>switch(config)# **userpassphrase min-length 8 max-length 80** | Restricts the minimum and/or maximum length of the user password. The minimum password length is from 4 to 127 characters, and the maximum password length is from 80 to 127 characters. |
| **Step 3** | (Optional) **show userpassphrase** {**length** \| **max-length** \| **min-length**}<br><br>**Example:**<br><br>switch(config)# **show userpassphrase length** | Displays the minimum and maximum length of the user password. |

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

## Enabling the Password Prompt for the Username

You can configure the switch to prompt the user to enter a password after entering the username.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal` | Enters global configuration mode. |
| **Step 2** | **password prompt username**<br><br>**Example:**<br><br>`switch(config)# password prompt username`<br>`Password prompt username is enabled.`<br>`After providing the required options in`<br>`the username command, press enter.`<br>`User will be prompted for the username`<br>`password and password will be hidden.`<br>`Note: Choosing password key in the same`<br>`line while configuring user account,`<br>`password will not be hidden.` | Configures the switch to prompt the user to enter a password after she enters the **username** command without the **password** option or the **snmp-server user** command. The password that the user enters will be hidden. You can use the **no** form of this command to disable this feature. |
| **Step 3** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

## Configuring the Shared Secret for RADIUS or TACACS+

The shared secret that you configure for remote authentication and accounting between the switch and the RADIUS or TACACS+ server should be hidden because it is sensitive information. You can use a separate command to generate an encrypted shared secret for the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands. The SHA256 hashing method is used to store the encrypted shared secret.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **generate type7_encrypted_secret**<br><br>**Example:**<br>`switch(config)# generate`<br>`type7_encrypted_secret`<br>`Type-7 (Vigenere) Encryption,`<br>` Use this encrypted secret to configure`<br>` radius and tacacs shared secret with`<br>`key type 7.`<br>` Copy complete secret with double quotes.`<br><br>`Enter plain text secret:`<br>`Confirm plain text secret:`<br>`Type 7 Encrypted secret is : "fewhg"` | Configures the RADIUS or TACACS+ shared secret with key type 7. You are prompted to enter the shared secret in plain text twice. The secret is hidden as you enter it. Then an encrypted version of the secret appears.<br><br>**Note** You can generate the encrypted equivalent of a plain-text secret separately and configure the encrypted shared secret later using the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands. |
| **Step 3** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Monitoring and Clearing the Local AAA Accounting Log

The Cisco NX-OS device maintains a local log for the AAA accounting activity. You can monitor this log and clear it.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **show accounting log** [*size* \| **last-index** \| **start-seqnum** *number* \| **start-time** *year month day hh:mm:ss*]<br><br>**Example:**<br>`switch# show accounting log` | 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 starting sequence number or a starting time for the log output.The range of the starting index is from 1 to 1000000. Use the **last-index** keyword to display the value of the last index number in the accounting log file. |
| **Step 2** | (Optional) **clear accounting log** [**logflash**]<br><br>**Example:**<br>`switch# clear aaa accounting log` | Clears the accounting log contents. The **logflash** keyword clears the accounting log stored in the logflash. |

# 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** {**ascii-authentication** | **chap** | **error-enable** | **mschap** | **mschapv2**}] | Displays AAA authentication login configuration information. |
| **show aaa groups** | Displays the AAA server group configuration. |
| **show login** [**failures**] | Displays the login parameters. The **failures** option displays information related only to failed login attempts.<br><br>**Note**  The **clear login failures** command clears the login failures in the current watch period. |
| **show login on-failure log** | Displays whether the switch is configured to log failed authentication messages to the syslog server. |
| **show login on-successful log** | Displays whether the switch is configured to log successful authentication messages to the syslog server. |
| **show running-config aaa** [**all**] | Displays the AAA configuration in the running configuration. |
| **show running-config all | i max-login** | Displays the maximum number of login sessions allowed per user. |
| **show startup-config aaa** | Displays the AAA configuration in the startup configuration. |
| **show userpassphrase** {**length** | **max-length** | **min-length**} | Displays the minimum and maximum length of the user password. |
| **show userpassphrase sequence alphabet length** | Displays the maximum alphabet sequence length of the user password. |
| **show userpassphrase sequence keyboard length** | Displays the maximum sequence keyboard length of the user password. |

# Configuration Examples for AAA

The following example shows how to configure AAA:

```
aaa authentication login default group radius
aaa authentication login console group radius
aaa accounting default group radius
```

# Configuration Examples for Login Parameters

The following example shows how to configure the switch to enter a 100-second quiet period if 3 failed login attempts is exceeded within 60 seconds. This example shows no login failures.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# 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 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Normal-Mode.
 Current Watch Window remaining time 45 seconds.
 Present login failure count 0.

switch(config)# show login failures
*** No logged failed login attempts with the device.***
```

The following example shows how to configure a quiet-mode ACL. All login requests are denied during the quiet period except hosts from the myacl ACL. This example also shows a login failure.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# login quiet-mode access-class myacl

switch(config)# show login

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Quiet-Mode.
 Will remain in Quiet-Mode for 98 seconds.
 Denying logins from all sources.

switch(config)# show login failures
Information about last 20 login failure's with the device.
--------------------------------------------------------------------------------
Username      Line          SourceIPAddr      Appname     TimeStamp
--------------------------------------------------------------------------------
asd           /dev/pts/0    171.70.55.158     login       Mon Aug  3 18:18:54 2015
qweq          /dev/pts/0    171.70.55.158     login       Mon Aug  3 18:19:02 2015
qwe           /dev/pts/0    171.70.55.158     login       Mon Aug  3 18:19:08 2015
```

--------------------------------------------------------------------------------

# Configuration Examples for the Password Prompt Feature

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **username** command and the error message that displays if she does not enter a password.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


switch(config)# username user1
Enter password:
Confirm password:
warning: password for user:user1 not set. S/he may not be able to login
```

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **snmp-server user** command and the prompts that then display to the user.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


N9K-1(config)# snmp-server user user1
Enter auth md5 password (Press Enter to Skip):
Enter auth sha password (Press Enter to Skip):
```

# Additional References for AAA

This section includes additional information related to implementing AAA.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS Licensing | *Cisco NX-OS Licensing Guide* |

**Standards**

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

**MIBs**

| MIBs | MIBs Link |
|---|---|
| MIBs related to AAA | To locate and download supported MIBs, go to the following URL: <br> ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

**CHAPTER 5**

# Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

## About RADIUS

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

## RADIUS Network Environments

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

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

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.

- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.

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

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

# RADIUS Operation

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

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

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

- 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 a delay in processing AAA requests. You can configure the Cisco NX-OS device to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive RADIUS servers as dead

and does not send AAA requests to any dead RADIUS servers. The Cisco NX-OS device periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco NX-OS device displays an error message that a failure is taking place.

**Figure 2: RADIUS Server States**

This figure shows the states for RADIUS server monitoring.



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

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

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

When you use RADIUS servers for authentication on a Cisco NX-OS 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 NX-OS software:

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

The Cisco NX-OS software 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 space. For example, if the user belongs to roles network-operator and network-admin, the value field would be network-operator network-admin. This subattribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute that is supported by the Cisco Access Control Server (ACS):

```
shell:roles=network-operator network-admin
```

```
shell:roles*"network-operator network-admin
```

The following examples show the roles attribute that is supported by FreeRADIUS:

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

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

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

**accountinginfo**

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

# About RADIUS Change of Authorization

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

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

The following requests are supported:

- Session reauthentication
- Session termination

# Session Reauthentication

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

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

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

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

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

# Session Termination

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

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

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

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

# Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.

- Obtain keys from the RADIUS servers.

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

# Guidelines and Limitations for RADIUS

RADIUS has the following guidelines and limitations:

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

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

- Only the RADIUS protocol supports one-time passwords.

- For N9K-X9636C-R and N9K-X9636Q-R line cards and the N9K-C9508-FM-R fabric module, RADIUS authentication fails for usernames with special characters.

- Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

- Beginning with Cisco NX-OS Release 10.3(1)F, RADIUS is supported on the Cisco Nexus 9808 platform switches.

    - Beginning with Cisco NX-OS Release 10.4(1)F, RADIUS is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, RADIUS is supported on the Cisco Nexus 9804 switches, X98900CD-A, and X9836DM-A line cards.

# Guidelines and Limitations for RadSec

RadSec has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.3(1)F, the RADIUS Secure (RadSec) support is provided on Cisco Nexus switches to secure the communication between RADIUS/TCP peers at the transport layer.

- RadSec must be enabled/disabled at the switch level, as the combination of servers having different transport protocols (i.e. UDP and TCP-with-TLS) is not possible.

- **radius-server directed-request** command is not supported along with the RadSec feature.

- **test aaa server radius** command is not supported for the RadSec servers, only **test aaa group** command is supported with the RadSec.

- Dot1x is not officially supported with RadSec.

- RADIUS server monitoring is not supported along with the RadSec servers.

- RADIUS server re-transmit and timeout are applicable to UDP based RADIUS mode and not supported for RadSec servers.

- Beginning with Cisco NX-OS Release 10.4(3)F, TLS version 1.3 and 1.2 is supported on Cisco Nexus switches. TLS v1.1 is deprecated.

# Guidelines and Limitations for RADIUS Change of Authorization

RADIUS Change of Authorization has the following guidelines and limitations:

- RADIUS Change of Authorization is supported on FEX.

• RADIUS change of Authorization is supported for VXLAN EVPN.

# Default Settings for RADIUS

This table lists the default settings for RADIUS parameters.

*Table 8: Default RADIUS Parameter Settings*

| Parameters | Default |
| --- | --- |
| Server roles | Authentication and accounting |
| Dead timer interval | 0 minutes |
| Retransmission count | 1 |
| Retransmission timer interval | 5 seconds |
| Authentication port | 1812 |
| Accounting port | 1813 |
| Idle timer interval | 0 minutes |
| Periodic server monitoring username | test |
| Periodic server monitoring password | test |

# Configuring RADIUS Servers

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

**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.

**Note** Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

# RADIUS Server Configuration Process

1. Establish the RADIUS server connections to the Cisco NX-OS device.

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

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

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

- Dead-time interval

- RADIUS server specification allowed at user login

- Timeout interval

- TCP port

**5.** (Optional) If RADIUS distribution is enabled, commit the RADIUS configuration to the fabric.

**Related Topics**

# Configuring RADIUS Server Hosts

To access a remote RADIUS server, you must configure the IP address or hostname of a RADIUS server. You can configure up to 64 RADIUS servers.

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

**Before you begin**

Ensure that the server is already configured as a member of the server group.

Ensure that the server is configured to authenticate RADIUS traffic.

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **radius-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.1.1** | Specifies the IPv4 or IPv6 address or hostname for a RADIUS server to use for authentication. |

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

**Related Topics**

# Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the Cisco NX-OS device. A RADIUS key is a shared secret text string between the Cisco NX-OS device and the RADIUS server hosts.

**Before you begin**

Obtain the RADIUS key values for the remote RADIUS servers.

Configure the RADIUS key on the remote RADIUS servers.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **radius-server key** [**0** \| **6** \| **7**] *key-value*<br><br>**Example:**<br><br>`switch(config)# radius-server key 0`<br>`QsEfThUkO`<br><br>**Example:**<br><br>`switch(config)# radius-server key 7`<br>`"fewhg"` | Specifies a RADIUS key for all RADIUS servers. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.<br><br>By default, no RADIUS key is configured. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | **Note** If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configuring the Shared Secret for RADIUS or TACACS+, on page 46. |
| **Step 3** | **exit** **Example:** `switch(config)# exit` `switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show radius-server** **Example:** `switch# show radius-server` | Displays the RADIUS server configuration. **Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys. |
| **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**

Configuring RADIUS Server Groups, on page 67
About AES Password Encryption and Primary Encryption Keys, on page 515

# Configuring a Key for a Specific RADIUS Server

You can configure a key on the Cisco NX-OS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the Cisco NX-OS device and a specific RADIUS server.

**Before you begin**

Configure one or more RADIUS server hosts.

Obtain the key value for the remote RADIUS server.

Configure the key on the RADIUS server.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **radius-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **key** [**0** \| **6** \| **7**] *key-value*<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.1.1 key 0 PlIjUhYg**<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.1.1 key 7 "fewhg"** | Specifies a RADIUS key for a specific RADIUS server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.<br><br>This RADIUS key is used instead of the global RADIUS key.<br><br>**Note** If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configuring the Shared Secret for RADIUS or TACACS+, on page 46. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show radius-server**<br><br>**Example:**<br><br>switch# **show radius-server** | Displays the RADIUS server configuration.<br><br>**Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

Configuring RADIUS Server Hosts, on page 60

About AES Password Encryption and Primary Encryption Keys, on page 515

# Configuring RadSec

RadSec is a protocol for transporting RADIUS datagrams over TLS.

This procedure describes how to enable/disable the RadSec on a switch.

### Before you begin

- Ensure that the client identity certificate and CA certificate of the server are installed on the switch.

- Ensure that the subject name in the server certificate is matching with the server host name/IP address that is configured on the switch.

- Before configuring AAA authentication and accounting to use RadSec servers, use **test aaa group** command and ensure RadSec authentication is success.

- Configure TLS idle-timeout to maximum value on RadSec server to avoid frequent TLS sessions retries from switch.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | **radius-server secure tls**<br><br>**Example:**<br><br>switch# **radius-server secure tls** | Enables the RadSec at global level.<br><br>**Note**    This CLI will not change or affect the port numbers that is used for RadSec. |
| **Step 3** | **radius-server host t** {*ipv4-address* \| *ipv6-address*\| *hostname*} **key** {*key*} **auth-port 2083 acct-port 2083 authentication accounting**<br><br>**Example:**<br><br>switch# **radius-server host 10.105.222.161 key radsec auth-port 2083 acct-port 2083 authentication accounting** | Configures the RadSec server with shared secret key along with the authentication and accounting ports.<br><br>**Note**    For server, the default RadSec port for authentication and accounting is "2083" and the key is "radsec". For switch, there is no default configuration for RadSec port and key, please add this configuration explicitly as defined on server. |
| **Step 4** | **radius-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **tls client-trustpoint** *trustpoint*<br><br>**Example:**<br><br>switch# **radius-server host 10.105.222.161 tls client-trustpoint rad1** | Configures the TLS client trustpoint where the client identity certificate is installed. |
| **Step 5** | **radius-server host** {*ipv4-address* \| *ipv6-address*\| *hostname*} **tls idle-timeout** *value* | Configures the TLS idle-timeout. The default value is 600 seconds. |

| Command or Action | Purpose |
|---|---|
| **Example:**<br><br>`switch# radius-server host 10.105.222.161`<br>`tls idle-timeout 80` | **Note**  If there are no transactions from the RadSec client, server can close the connection based on its timeout value. The TLS idle-timeout on the client is not supported in this release. Client does not close connections on its own. |

**Note**  When remote user logs-in, you can notice delay in login for approximately 20 seconds i.e when TLS session establishment is happening for the first time between switch and RadSec server, Once TLS sessions are up no delay will be seen for consecutive remote log-ins.

**Note**  When a RadSec client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, you may experience delay in show run commands.

# About RADIUS with DTLS

From Cisco NX-OS Release 10.4(1)F, RADIUS with DTLS protocol is introduced. This protocol is for transporting RADIUS datagrams over a secure channel using UDP.

RADIUS with DTLS provides secure communication between RADIUS peers at the transport layer. This protocol helps secure RADIUS packets transfer through different administrative domains and suspicious, and unsafe networks.

# Configuring RADIUS with DTLS

### Before you begin

- Ensure that you create client identity certificate with subject and alternative name same as the IP address/DNS hostname of the switch. Install the client identity certificate on the switch using a trustpoint.

- Ensure that the server certificate of ISE server used for DTLS/RADIUS is installed on the switch.

- Make sure that the CA certificate used to sign client identity certificate is installed in trusted certificate store of ISE server.

- Ensure that the subject name in the server certificate is same as the server hostname/IP address that is configured on the switch.

- Before configuring AAA authentication and accounting groups to use RADIUS servers, check with test aaa group command and ensure that the RADIUS authentication is successful.

- You must enable RADIUS with DTLS protocol at the switch level.

- Configuring combination of RADIUS servers to use different transports protocols such as DTLS and TLS is not supported. You can configure one protocol at an instant.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br><br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **radius-server secure dtls**<br><br>**Example:**<br><br>`switch(config)# radius-server secure dtls` | Enables the RADIUS with DTLS protocol on the switch. |
| **Step 3** | **radius-server host** {*ipv4-address \| ipv6-address\| hostname*} **key** {*radius/dtls*} **auth-port 2083 acct-port 2083 authentication accounting**<br><br>**Example:**<br><br>`switch(config)# radius-server host 10.105.222.161 key radius/dtls auth-port 2083 acct-port 2083 authentication accounting` | Configures the RADIUS server with shared secret key along with the authentication and accounting ports.<br><br>**Note** The default destination DTLS port for authentication and accounting is **UDP/2083**. There is no default server key for DTLS as per RFC. Ensure that you add this configuration explicitly as defined on server. The ISE server must be pre-set with the "radius/dtls" key at that instant. Check and add the key on the Nexus switch while configuring DTLS with an ISE server. |
| **Step 4** | **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **dtls client-trustpoint** *trustpoint*<br><br>**Example:**<br><br>`switch(config)# radius-server host 10.105.222.161 dtls client-trustpoint rad1` | Configures the DTLS client-trustpoint parameter with a trustpoint where the switch identity certificate is installed. The *rad1* is a trustpoint on the switch which must have the client identity certificate. |
| **Step 5** | **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **dtls idle-timeout** *value*<br><br>**Example:**<br><br>`switch# radius-server host 10.105.222.161 dtls idle-timeout 80` | Configures the DTLS idle-timeout. The default value is 600 seconds.<br><br>**Note** If there are no transactions from the RADIUS client, server can close the connection as per defined timeout value. The DTLS idle-timeout on the client is not supported in this release. Client does not close connections on its own. |

**Note**

- When a remote user logs in, you may notice a delay of approximately 20 seconds, which occurs when the TLS session is being established for the first time between the switch and the RADIUS server. Once the TLS sessions are up, no delay will be seen for consecutive remote logins.

- When a RADIUS client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, you may experience delay in the `show run` commands.

# 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.

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**

Ensure that all servers in the group are RADIUS servers.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **aaa group server radius** *group-name*<br><br>**Example:**<br><br>`switch(config)# aaa group server radius RadServer`<br>`switch(config-radius)#` | Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.<br><br>To delete a RADIUS server group, use the **no** form of this command.<br><br>**Note** You are not allowed to delete the default system generated default group (RADIUS). |
| **Step 3** | **server** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br><br>`switch(config-radius)# server 10.10.1.1` | Configures the RADIUS server as a member of the RADIUS server group.<br><br>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) **deadtime** *minutes*<br><br>**Example:**<br>switch(config-radius)# **deadtime 30** | Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440.<br><br>**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) **server** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br>switch(config-radius)# **server 10.10.1.1** | Configures the RADIUS server as a member of the RADIUS server group.<br><br>**Tip**    If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command. |
| **Step 6** | | (Optional) **use-vrf** *vrf-name*<br><br>**Example:**<br>switch(config-radius)# **use-vrf vrf1** | Specifies the VRF to use to contact the servers in the server group. |
| **Step 7** | | **exit**<br><br>**Example:**<br>switch(config-radius)# **exit**<br>switch(config)# | Exits configuration mode. |
| **Step 8** | | (Optional) **show radius-server groups** [*group-name*]<br><br>**Example:**<br>switch(config)# **show radius-server groups** | Displays the RADIUS server group configuration. |
| **Step 9** | | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

# 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. By default, the Cisco NX-OS software uses any available interface.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br><br> **Example:** <br><br> switch# **configure terminal** <br> switch(config) | Enters global configuration mode. |
| **Step 2** | **ip radius source-interface** *interface* <br><br> **Example:** <br><br> switch(config)# **ip radius** <br> **source-interface mgmt 0** | Configures the global source interface for all RADIUS server groups configured on the device. |
| **Step 3** | **exit** <br><br> **Example:** <br><br> switch(config)# **exit** <br> switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show radius-server** <br><br> **Example:** <br><br> switch# **show radius-server** | Displays the RADIUS server configuration information. |
| **Step 5** | (Optional) **copy running-config startup config** <br><br> **Example:** <br><br> switch# **copy running-config** <br> **startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

# Allowing Users to Specify a RADIUS Server at Login

By default, the Cisco NX-OS device forwards an authentication request based on the default AAA authentication method. You can configure the Cisco NX-OS device to allow the user to specify a VRF and RADIUS server to send the authentication request by enabling the directed-request option. If you enable this option, the user can log in as *username*@*vrfname***:***hostname*, where *vrfname* is the VRF to use and **hostname** is the name of a configured RADIUS server.

**Note** If you enable the directed-request option, the Cisco NX-OS device uses only the RADIUS method for authentication and not the default local method.

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **radius-server directed-request**<br><br>**Example:**<br><br>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** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show radius-server directed-request**<br><br>**Example:**<br><br>switch# **show radius-server directed-request** | Displays the directed request configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

# 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 Cisco NX-OS device 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 NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **radius-server retransmit** *count*<br><br>**Example:**<br><br>`switch(config)# `**`radius-server retransmit`**<br>**`3`** | Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5. |
| Step 3 | **radius-server timeout** *seconds*<br><br>**Example:**<br><br>`switch(config)# radius-server timeout 10` | 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 | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| Step 5 | (Optional) **show radius-server**<br><br>**Example:**<br><br>`switch# `**`show radius-server`** | Displays the RADIUS server configuration. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>**`startup-config`** | Copies the running configuration to the startup configuration. |

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

By default, a Cisco NX-OS device retries a 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 Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

**Before you begin**

Configure one or more RADIUS server hosts.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **radius-server host** {*ipv4-address* \|<br>*ipv6-address* \| *hostname*} **retransmit** *count* | Specifies the retransmission count for a specific server. The default is the global value. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>switch(config)# **radius-server host server1 retransmit 3** | **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers. |
| **Step 3** | **radius-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **timeout** *seconds*<br>**Example:**<br>switch(config)# **radius-server host server1 timeout 10** | Specifies the transmission timeout interval for a specific server. The default is the global value.<br><br>**Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers. |
| **Step 4** | **exit**<br>**Example:**<br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 5** | (Optional) **show radius-server**<br>**Example:**<br>switch# **show radius-server** | Displays the RADIUS server configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

# 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 if there is a conflict with the default port.

### Before you begin

Configure one or more RADIUS server hosts.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:** | Enters global configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | switch# **configure terminal**<br>switch(config)# | |
| **Step 2** | (Optional) **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **acct-port** *udp-port*<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.1.1 acct-port 2004** | Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1813. The range is from 0 to 65535. |
| **Step 3** | (Optional) **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **accounting**<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.1.1 accounting** | Specifies to use the RADIUS server only for accounting purposes. The default is both accounting and authentication. |
| **Step 4** | (Optional) **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **auth-port** *udp-port*<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.2.2 auth-port 2005** | 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) **radius-server host** {*ipv4-address \| ipv6-address \| hostname*} **authentication**<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.10.2.2 authentication** | Specifies to use the RADIUS server only for authentication purposes. The default is both accounting and authentication. |
| **Step 6** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 7** | (Optional) **show radius-server**<br><br>**Example:**<br><br>switch# **show radius-server** | Displays the RADIUS server configuration. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

Configuring RADIUS Server Hosts, on page 60

# Configuring Global Periodic RADIUS Server Monitoring

You can monitor the availability of all RADIUS servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

> **Note** Test parameters that are configured for individual servers take precedence over global test parameters.

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

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

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

### Before you begin

Enable RADIUS.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal```<br>```switch(config)#``` | Enters global configuration mode. |
| **Step 2** | **radius-server test** {**idle-time** *minutes* \| **password** *password* [**idle-time** *minutes*] \| **username** *name* [**password** *password* [**idle-time** *minutes*]]}<br><br>**Example:**<br><br>```switch(config)# radius-server test```<br>```username user1 password Ur2Gd2BH```<br>```idle-time 3``` | Specifies parameters for global server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.<br><br>**Note** For periodic RADIUS server monitoring, the idle timer value must be greater than 0. |
| **Step 3** | **radius-server deadtime** *minutes*<br><br>**Example:** | Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# **radius-server deadtime 5** | value is 0 minutes, and the valid range is from 0 to 1440 minutes. |
| **Step 4** | **exit**<br><br>**Example:**<br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 5** | (Optional) **show radius-server**<br><br>**Example:**<br>switch# **show radius-server** | Displays the RADIUS server configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Periodic RADIUS Server Monitoring on Individual Servers

You can monitor the availability of individual RADIUS servers. The configuration 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 Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

> **Note** Test parameters that are configured for individual servers take precedence over global test parameters.

> **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.

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

**Before you begin**

Enable RADIUS.

Add one or more RADIUS server hosts.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **radius-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **test** {**idle-time** *minutes* \| **password** *password* [**idle-time** *minutes*] \| **username** *name* [**password** *password* [**idle-time** *minutes*]]}<br><br>**Example:**<br>`switch(config)# radius-server host`<br>`10.10.1.1 test username user1 password`<br>`Ur2Gd2BH idle-time 3` | Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.<br><br>**Note**    For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0. |
| **Step 3** | **radius-server deadtime** *minutes*<br><br>**Example:**<br>`switch(config)# radius-server deadtime`<br>`5` | Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 1440 minutes. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show radius-server**<br><br>**Example:**<br>`switch# show radius-server` | Displays the RADIUS server configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring the RADIUS Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco NX-OS 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 icon]

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

**Related Topics**

# Configuring One-Time Passwords

One-time password (OTP) support is available for Cisco NX-OS devices through the use of RSA SecurID token servers. With this feature, users authenticate to a Cisco NX-OS device by entering both a personal identification number (or one-time password) and the token code being displayed at that moment on their RSA SecurID token.

![Note icon]

**Note** The token code used for logging into the Cisco NX-OS device changes every 60 seconds. To prevent problems with device discovery, we recommend using different usernames that are present on the Cisco Secure ACS internal database.

**Before you begin**

On the Cisco NX-OS device, configure a RADIUS server host and remote default login authentication.

Ensure that the following are installed:

- Cisco Secure Access Control Server (ACS) version 4.2

- RSA Authentication Manager version 7.1 (the RSA SecurID token server)

- RSA ACE Agent/Client

No configuration (other than a RADIUS server host and remote authentication) is required on the Cisco NX-OS device to support one-time passwords. However, you must configure the Cisco Secure ACS as follows:

1. Enable RSA SecurID token server authentication.

2. Add the RSA SecurID token server to the Unknown User Policy database.

# Manually Monitoring RADIUS Servers or Groups

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 1 | **test aaa server radius** {*ipv4-address* \| *ipv6-address* \| *hostname*} [**vrf** *vrf-name*] *username password*<br><br>**Example:**<br><br>`switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH` | Sends a test message to a RADIUS server to confirm availability. |
| Step 2 | **test aaa group** *group-name username password*<br><br>**Example:**<br><br>`switch# test aaa group RadGroup user2 As3He3CI` | Sends a test message to a RADIUS server group to confirm availability. |

# Enabling or Disabling Dynamic Author Server

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **aaa server radius dynamic-author** <br><br> **Example:** <br><br> switch(config)# **aaa server radius dynamic-author** | Enables the RADIUS dynamic author server. You can disable the RADIUS dynamic author server using the no form of this command. |

# Configuring RADIUS Change of Authorization

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal** <br><br> **Example:** <br><br> switch# **configure terminal** <br> switch(config)# | Enters global configuration mode. |
| **Step 2** | [**no**] **aaa server radius dynamic-author** <br><br> **Example:** <br><br> switch(config)# **aaa server radius dynamic-author** | Configures the switch as an AAA server to facilitate interaction with an external policy server. You can disable the RADIUS dynamic author and the associated clients using the no form of this command. |
| **Step 3** | [**no**] **client** {*ip-address* \| **hostname** } [**server-key** [**0** \| **7** ] *string* ] <br><br> **Example:** <br><br> switch(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1 | Configures the IP address or the hostname of the AAA server client. Use the optional server-key keyword and string argument to configure the server key at the client level. You can remove the client server using the no form of this command. <br><br> **Note**    Configuring the server key at the client level overrides the server key that is configured at the global level. |
| **Step 4** | [**no**] **port** *port-number* <br><br> **Example:** <br><br> switch(config-locsvr-da-radius)# port 3799 | Specifies the port on which a device listens to the RADIUS requests from the configured RADIUS clients. The port range is 1 - 65535. You can revert to the default port using the no form of this command. <br><br> **Note**    The default port for a packet of disconnect is 1700. |
| **Step 5** | [**no**] **server-key** [**0** \| **7** ] *string* | Configures the global RADIUS key to be shared between a device and the RADIUS clients. You can remove the server-key using the no form of this command. |

# Verifying the RADIUS Configuration

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

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

# Verifying RADIUS Change of Authorization Configuration

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

| Command | Purpose |
|---------|---------|
| **show running-config dot1x** | Displays the dot1x configuration in the running configuration. |
| **show running-config aaa** | Displays the AAA configuration in the running configuration. |
| **show running-config radius** | Displays the RADIUS configuration in the running configuration. |
| **show aaa server radius statistics** | Displays the local RADIUS server statistics. |
| **show aaa client radius statistics** {*ip address* \| *hostname* } | Displays the local RADIUS client statistics. |
| **clear aaa server radius statistics** | Clears the local RADIUS server statistics. |
| **clear aaa client radius statistics** {*ip address* \| *hostname* } | Clears the local RADIUS client statistics. |

# Monitoring RADIUS Servers

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

**Before you begin**

Configure one or more RADIUS server hosts.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **show radius-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>switch# **show radius-server statistics 10.10.1.1** | Displays the RADIUS statistics. |

**Related Topics**

# 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) **show radius-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>switch# **show radius-server statistics 10.10.1.1** | Displays the RADIUS server statistics on the Cisco NX-OS device. |
| **Step 2** | **clear radius-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>switch# **clear radius-server statistics 10.10.1.1** | Clears the RADIUS server statistics. |

**Related Topics**

# Configuration Example for RADIUS

The following example shows how to configure RADIUS:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
aaa group server radius RadServer
    server 10.10.1.1
```

# Configuration Examples of RADIUS Change of Authorization

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

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

# Where to Go Next

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

# Additional References for RADIUS

This section describes additional information related to implementing RADIUS.

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS Licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

### Standards

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

### MIBs

| MIBs | MIBs Link |
|---|---|
| MIBs related to RADIUS | To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

# Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

## About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to a Cisco NX-OS device. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your Cisco NX-OS 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 can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. Cisco NX-OS devices provide centralized authentication using the TACACS+ protocol.

Beginning with Cisco NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the **aaa authorization ssh-certificate default group** command on the Cisco Nexus 9000 Series platform switches. For configuration details, see Configuring X.509 Certificate-Based SSH Authorization Using TACACS Server, on page 107

# TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco NX-OS 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.

# TACACS+ Operation for User Login

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

**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 your mother's maiden name.

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

2. The Cisco NX-OS device will eventually receive one of the following responses from the TACACS+ daemon:

   **ACCEPT**
   User authentication succeeds and service begins. If the Cisco NX-OS 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 either at the daemon or in the network connection between the daemon and the Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS device tries to use an alternative method for authenticating the user.

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

3. If TACACS+ authorization is required, the Cisco NX-OS 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 or IPv6), access list, and user timeouts

# Default TACACS+ Server Encryption Type and Secret Key

You must configure the TACACS+ secret key to authenticate the switch to the TACACS+ server. A secret key is a secret text string shared between the Cisco NX-OS 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 secret key for all TACACS+ server configurations on the Cisco NX-OS device to use.

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

# Command Authorization Support for TACACS+ Servers

By default, command authorization is done against a local database in the Cisco NX-OS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

# TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor a TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A Cisco NX-OS 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 its way. Whenever a TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance.

**Figure 3: TACACS+ Server States**

This figure shows the server states for TACACS+ server monitoring.

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

# Vendor-Specific Attributes for TACACS+

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

## Cisco VSA Format for TACACS+

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

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

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

When you use TACACS+ servers for authentication on a Cisco NX-OS device, the TACACS+ protocol directs the TACACS+ server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Cisco NX-OS software:

**Shell**

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

**Accounting**

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

The Cisco NX-OS software 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 space. For example, if the user belongs to roles network-operator and network-admin, the value field would be network-operator network-admin. This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Cisco ACS:

```
shell:roles=network-operator network-admin
```

```
shell:roles*network-operator network-admin
```

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

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

# Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.

- Obtain the secret keys from the TACACS+ servers, if any.

- Ensure that the Cisco NX-OS device is configured as a TACACS+ client of the AAA servers.

# Guidelines and Limitations for TACACS+

TACACS+ has the following guidelines and limitations:

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

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

- Cisco recommends that you configure the dead-time interval if more than six servers are configured in a group. If you must configure more than six servers, make sure to set the dead-time interval to a value greater than 0 and enable dead server monitoring by configuring the test username and test password.

- Command authorization on TACACS+ servers is available for console sessions.

- For N9K-X9636C-R and N9K-X9636Q-R line cards and the N9K-C9508-FM-R fabric module, TACACS+ authentication fails for usernames with special characters.

- Beginning with Cisco NX-OS Release 10.3(1)F, TACACS+ is supported on the Cisco Nexus 9808 switches.

    - Beginning with Cisco NX-OS Release 10.4(1)F, TACACS+ is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, TACACS+ is supported on the Cisco Nexus 9804 switches,X98900CD-A and X9836DM-A line cards.

- Beginning with Cisco NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the aaa authorization ssh-certificate default group command on the Cisco Nexus 9000 Series platform switches.

- The Cisco NX-OS switches do not support custom username/password prompts. If custom prompts are provided to the switch, they will be ignored.

# Default Settings for TACACS+

This table lists the default settings for TACACS+ parameters.

**Table 9: Default TACACS+ Parameters Settings**

| Parameters | Default |
|---|---|
| TACACS+ | Disabled |
| Dead timer interval | 0 minutes |
| Timeout interval | 5 seconds |
| Idle timer interval | 0 minutes |
| Periodic server monitoring username | test |
| Periodic server monitoring password | test |
| Privilege level support for TACACS+ authorization | Disabled |

# One-Time Password Support

A one-time password (OTP) is a password that is valid for a single login session or a transaction. OTPs avoid multiple disadvantages that are associated with the static passwords. OTPs are not at risk to replay attacks. If an intruder manages to record an OTP that was already used to log into a service or to conduct an operation, it cannot be misused because it is no longer valid.

OTPs are applicable only to the RADIUS and TACACS+ protocol daemons. For a RADIUS protocol daemon, you must ensure that you disable the ASCII authentication mode. For a TACACS+ protocol daemon, you must enable the ASCII authentication mode. To enable the ASCII authentication mode, use the **aaa authentication login ascii-authentication** command.

# Configuring TACACS+

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

**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.

# TACACS+ Server Configuration Process

**Procedure**

| | | |
|---|---|---|
| **Step 1** | Enable TACACS+. |
| **Step 2** | Establish the TACACS+ server connections to the Cisco NX-OS device. |
| **Step 3** | Configure the 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** | (Optional) Configure the TCP port. |
| **Step 6** | (Optional) If needed, configure periodic TACACS+ server monitoring. |
| **Step 7** | (Optional) If TACACS+ distribution is enabled, commit the TACACS+ configuration to the fabric. |

**Related Topics**

Enabling TACACS+ , on page 89

# Enabling TACACS+

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

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **feature tacacs+**<br><br>**Example:**<br><br>`switch(config)# feature tacacs+` | Enables TACACS+. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`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 IP address or the hostname for the TACACS+ server on the Cisco NX-OS device. You can configure up to 64 TACACS+ servers.

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

### Before you begin

Enable TACACS+.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote TACACS+ servers.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **tacacs-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br><br>`switch(config)# `**`tacacs-server host`**<br>**`10.10.2.2`** | Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server. |
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>`switch(config)# `**`show tacacs+ pending`** | Displays the TACACS+ configuration pending for distribution. |
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br><br>`switch(config)# `**`tacacs+ commit`** | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| **Step 6** | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>`switch# `**`show tacacs-server`** | Displays the TACACS+ server configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Global TACACS+ Keys

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

**Before you begin**

Enable TACACS+.

Obtain the secret key values for the remote TACACS+ servers.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **tacacs-server key** [**0** \| **6** \| **7**] *key-value*<br><br>**Example:**<br><br>`switch(config)# tacacs-server key 0`<br>`QsEfThUkO`<br><br>**Example:**<br><br>`switch(config)# tacacs-server key 7`<br>`"fewhg"` | Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.<br><br>By default, no secret key is configured.<br><br>**Note**  If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configuring the Shared Secret for RADIUS or TACACS+, on page 46. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **exit**<br><br>**Example:**<br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show tacacs-server**<br><br>**Example:**<br>`switch# `**`show tacacs-server`** | Displays the TACACS+ server configuration.<br><br>**Note** The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# `**`copy running-config`**<br>**`startup-config`** | Copies the running configuration to the startup configuration. |

**Related Topics**

Enabling TACACS+ , on page 89

About AES Password Encryption and Primary Encryption Keys, on page 515

# Configuring a Key for a Specific TACACS+ Server

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

### Before you begin

Enable TACACS+.

Obtain the secret key values for the remote TACACS+ servers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **tacacs-server host** {*ipv4-address* \| *ipv6-address* \| *host-name*} **key** [**0** \| **6** \| **7**] *key-value*<br><br>**Example:**<br>`switch(config)# `**`tacacs-server host`**<br>**`10.10.1.1 key 0 PlIjUhYg`**<br><br>**Example:** | Specifies a secret key for a specific TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# **tacacs-server host 10.10.1.1 key 7 "fewhg"** | format is clear text. The maximum length is 63 characters. |
| | | This secret key is used instead of the global secret key. |
| | | **Note**    If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configuring the Shared Secret for RADIUS or TACACS+, on page 46. |
| **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. |
| | | **Note**    The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys. |
| **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**

About AES Password Encryption and Primary Encryption Keys, on page 515

# 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**

Enable TACACS+.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal<br>switch(config)#``` | Enters global configuration mode. |
| **Step 2** | **aaa group server tacacs+** *group-name*<br><br>**Example:**<br><br>```switch(config)# aaa group server tacacs+ TacServer<br>switch(config-tacacs+)#``` | Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group. |
| **Step 3** | **server** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br><br>```switch(config-tacacs+)# server 10.10.2.2``` | Configures the TACACS+ server as a member of the TACACS+ server group.<br><br>If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>```switch(config-tacacs+)# exit<br>switch(config)#``` | Exits TACACS+ server group configuration mode. |
| **Step 5** | (Optional) **show tacacs-server groups**<br><br>**Example:**<br><br>```switch(config)# show tacacs-server groups``` | Displays the TACACS+ server group configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```switch(config)# copy running-config<br>startup-config``` | Copies the running configuration to the startup configuration. |

**Related Topics**

Enabling TACACS+ , on page 89

Remote AAA Services, on page 20

Configuring TACACS+ Server Hosts, on page 90

Configuring the TACACS+ Dead-Time Interval, on page 102

# 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. By default, the Cisco NX-OS software uses any available interface.

**Procedure**

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

**Related Topics**

Enabling TACACS+ , on page 89

Configuring TACACS+ Server Groups, on page 93

# Allowing Users to Specify a TACACS+ Server at Login

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

**Note** If you enable the directed-request option, the Cisco NX-OS device uses only the TACACS+ method for authentication and not the default local method.

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

**Before you begin**

Enable TACACS+.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **tacacs-server directed-request**<br><br>**Example:**<br><br>switch(config)# **tacacs-server**<br>**directed-request** | Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled. |
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>switch(config)# **show tacacs+ pending** | Displays the pending TACACS+ configuration. |
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br><br>switch(config)# **tacacs+ commit** | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 6** | (Optional) **show tacacs-server directed-request**<br><br>**Example:**<br><br>switch# **show tacacs-server**<br>**directed-request** | Displays the TACACS+ directed request configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config**<br>**startup-config** | Copies the running configuration to the startup configuration. |

# Configuring the Timeout Interval for a TACACS+ Server

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

**Before you begin**

Enable TACACS+.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **tacacs-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **timeout** *seconds*<br><br>**Example:**<br><br>switch(config)# **tacacs-server host server1 timeout 10** | Specifies the timeout interval for a specific server. The default is the global value.<br><br>**Note**   The timeout interval value specified for a TACACS+ server overrides the global timeout interval value specified for all TACACS+ servers. |
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>switch(config)# **show tacacs+ pending** | Displays the TACACS+ configuration pending for distribution. |
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br><br>switch(config)# **tacacs+ commit** | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 6** | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>switch# **show tacacs-server** | Displays the TACACS+ server configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, Cisco NX-OS devices use port 49 for all TACACS+ requests.

**Before you begin**

Enable TACACS+.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **tacacs-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **port** *tcp-port*<br><br>**Example:**<br><br>`switch(config)# tacacs-server host`<br>`10.10.1.1 port 2` | Specifies the TCP port to use for TACACS+ messages to the server. The default TCP port is 49. The range is from 1 to 65535. |
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>`switch(config)# show tacacs+ distribution`<br>` pending` | Displays the TACACS+ configuration pending for distribution. |
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br><br>`switch(config)# tacacs+ commit` | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 6** | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>`switch# show tacacs-server` | Displays the TACACS+ server configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Enabling TACACS+ , on page 89

# Configuring Global Periodic TACACS+ Server Monitoring

You can monitor the availability of all TACACS+ servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

**Note** Test parameters that are configured for individual servers take precedence over global test parameters.

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

**Note** The test parameters are distributed across switches. If even one switch in the fabric is running an older release, the test parameters are not distributed to any switch in the fabric.

**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.

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

**Before you begin**

Enable TACACS+.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **tacacs-server test** {**idle-time** *minutes* \| **password** *password* [**idle-time** *minutes*] \| **username** *name* [**password** *password* [**idle-time** *minutes*]]}<br><br>**Example:**<br><br>`switch(config)# tacacs-server test`<br>`username user1 password Ur2Gd2BH`<br>`idle-time 3` | Specifies parameters for global server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.<br><br>**Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0. |
| **Step 3** | **tacacs-server dead-time** *minutes*<br><br>**Example:**<br><br>`switch(config)# tacacs-server dead-time`<br>` 5` | Specifies the number of minutes before the Cisco NX-OS device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>`switch# show tacacs-server` | Displays the TACACS+ server configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Periodic TACACS+ Server Monitoring on Individual Servers

You can monitor the availability of individual TACACS+ servers. The configuration 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 NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

**Note**  Test parameters that are configured for individual servers take precedence over global test parameters.

**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.

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

**Note**  The test parameters are distributed across switches. The test parameters are not distributed to any switch in the fabric.

### Before you begin

Enable TACACS+.

Add one or more TACACS+ server hosts.

### Procedure

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br><br>**Example:** <br><br>switch# **configure terminal** <br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **tacacs-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **test** {**idle-time** *minutes* \| **password** *password* [**idle-time** *minutes*] \| **username** *name* [**password** *password* [**idle-time** *minutes*]]} <br><br>**Example:** <br><br>switch(config)# **tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3** | Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes. <br><br>**Note**  For periodic TACACS+ server monitoring, the idle timer value must be greater than 0. |
| **Step 3** | **tacacs-server dead-time** *minutes* <br><br>**Example:** <br><br>switch(config)# **tacacs-server dead-time 5** | Specifies the number of minutes before the Cisco NX-OS device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes. |

| | | Command or Action | Purpose |
|---|---|---|---|
| Step 4 | | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| Step 5 | | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>`switch# `**`show tacacs-server`** | Displays the TACACS+ server configuration. |
| Step 6 | | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>**`startup-config`** | Copies the running configuration to the startup configuration. |

**Related Topics**

Configuring TACACS+ Server Hosts, on page 90

Configuring Global Periodic TACACS+ Server Monitoring, on page 99

# Configuring the TACACS+ Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco NX-OS 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-timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per group.

**Before you begin**

Enable TACACS+.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **tacacs-server deadtime** *minutes*<br><br>**Example:**<br><br>`switch(config)# `**`tacacs-server deadtime`**<br>**`5`** | Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes. |

| | Command or Action | Purpose |
|---|---|---|
| Step 3 | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br>switch(config)# **show tacacs+ pending** | Displays the pending TACACS+ configuration. |
| Step 4 | (Optional) **tacacs+ commit**<br><br>**Example:**<br>switch(config)# **tacacs+ commit** | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| Step 5 | **exit**<br><br>**Example:**<br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| Step 6 | (Optional) **show tacacs-server**<br><br>**Example:**<br>switch# **show tacacs-server** | Displays the TACACS+ server configuration. |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>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**<br><br>**Example:**<br>switch# **configure terminal**<br>switch(config)# | Enters global configuration mode. |
| Step 2 | **aaa authentication login ascii-authentication**<br><br>**Example:**<br>switch(config)# **aaa authentication login ascii-authentication** | Enables ASCII authentication. The default is disabled. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>switch(config)# **show tacacs+ pending** | Displays the pending TACACS+ configuration. |
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br><br>switch(config)# **tacacs+ commit** | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>switch(config)# **exit**<br>switch# | Exits configuration mode. |
| **Step 6** | (Optional) **show tacacs-server**<br><br>**Example:**<br><br>switch# **show tacacs-server** | Displays the TACACS+ server configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

# Configuring Command Authorization on TACACS+ Servers

You can configure authorization for commands on TACACS+ servers.

⚠️

**Caution**  Command authorization disables user role-based authorization control (RBAC), including the default roles.

✏️

**Note**  If you use a console to login to the server, command authorization is disabled. Authorization is available for both non-console and console sessions. By default, command authorization is disabled for console sessions even if it is configured for default (non-console) sessions. You must explicitly configure a AAA group for the console to enable command authorization for console sessions.

✏️

**Note**  By default, context sensitive help and command tab completion show only the commands supported for a user as defined by the assigned roles. When you enable command authorization, the Cisco NX-OS software displays all commands in the context sensitive help and in tab completion, regardless of the role assigned to the user.

**Before you begin**

Enable TACACS+.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **aaa authorization** {**commands** \| **config-commands**} {**console** \| **default**} {**group** *group-list* [**local**] \| **local**}<br><br>**Example:**<br><br>`switch(config)# aaa authorization`<br>`commands`<br>`default group TacGroup`<br>`Per command authorization will disable`<br>`RBAC for all`<br>`users. Proceed (y/n)?` | Configures the command authorization method for specific roles on a TACACS+ server.<br><br>The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword configures authorization sources for all configuration commands.<br><br>The **console** keyword configures command authorization for a console session, and the **default** keyword configures command authorization for a non-console session.<br><br>The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers belonging to this group are contacted for command authorization. The **local** method uses the local role-based database for authorization.<br><br>The **local** method is used only if all the configured server groups fail to respond and you have configured **local** as the fallback method. The default method is **local**.<br><br>If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond.<br><br>If you press **Enter** at the confirmation prompt, the default action is **n**. |
| **Step 3** | (Optional) **show tacacs+** {**pending** \| **pending-diff**}<br><br>**Example:**<br><br>`switch(config)# show tacacs+ pending` | Displays the pending TACACS+ configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **tacacs+ commit**<br><br>**Example:**<br>switch(config)# tacacs+ commit | Applies the TACACS+ configuration changes in the temporary database to the running configuration. |
| **Step 5** | **exit**<br><br>**Example:**<br>switch(config)# exit<br>switch# | Exits global configuration mode. |
| **Step 6** | (Optional) **show aaa authorization** [**all**]<br><br>**Example:**<br>switch(config)# show aaa authorization | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

**Related Topics**

Enabling TACACS+ , on page 89

Testing Command Authorization on TACACS+ Servers, on page 106

# Testing Command Authorization on TACACS+ Servers

You can test the command authorization for a user on the TACACS+ servers.

> **Note** You must send correct commands for authorization or else the results may not be reliable.

> **Note** The **test** command uses the default (non-console) method for authorization, not the console method.

**Before you begin**

Enable TACACS+.

Ensure that you have configured command authorization for the TACACS+ servers.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **test aaa authorization command-type** {**commands** \| **config-commands**} **user** *username* **command** *command-string*<br><br>**Example:**<br>`switch# test aaa authorization`<br>`command-type commands`<br>`user TestUser command reload` | Tests a user's authorization for a command on the TACACS+ servers.<br><br>The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands.<br><br>**Note**    Put double quotes (") before and after the *command-string* argument if it contains spaces. |

**Related Topics**

# Enabling and Disabling Command Authorization Verification

You can enable and disable command authorization verification on the command-line interface (CLI) for the default user session or for another username.

> **Note**    The commands do no execute when you enable authorization verification.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **terminal verify-only** [**username** *username*]<br><br>**Example:**<br>`switch# terminal verify-only` | Enables command authorization verification. After you enter this command, the Cisco NX-OS software indicates whether the commands you enter are authorized or not. |
| **Step 2** | **terminal no verify-only** [**username** *username*]<br><br>**Example:**<br>`switch# terminal no verify-only` | Disables command authorization verification. |

# Configuring X.509 Certificate-Based SSH Authorization Using TACACS Server

Beginning with Cisco NX-OS release 10.4(3)F, you can configure SSH-based authorization of x509v3-certificates using a TACAC+ server on the Cisco Nexus switches.

To configure X.509 certificate-based SSH-authorization using a TACAC+ server, follow these steps:

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **aaa authorization ssh-certificate default group** *tacacs-group-name*<br><br>**Example:**<br><br>`switch(config)# `**`aaa authorization`**<br>**`ssh-certificate default group tac`** | Configures the default AAA authorization method for the TACAC+ servers.<br><br>The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.<br><br>**Note** • Ensure that the *tacacs-group-name* is configured under the TACACS-server configuration using the **aaa group server tacacs+** *tacacs-group-name* command.<br><br>• To support SSH certificate-based authentication, configure a crypto trustpoint and install the root CA. For more details, see the Configuring PKI, on page 167 section. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# `**`exit`**<br>`switch#` | Exits global configuration mode. |
| **Step 4** | (Optional) **show aaa authorization** [**all**]<br><br>**Example:**<br><br>`switch# show aaa authorization` | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# `**`copy running-config`**<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Permitting or Denying Commands for Users of Privilege Roles

As a network administrator, you can modify the privilege roles to permit users to execute specific commands or to prevent users from running those commands.

You must follow these guidelines when changing the rules of privilege roles:

- You cannot modify the priv-14 and priv-15 roles.

- You can add deny rules only to the priv-0 role.

- These commands are always permitted for the priv-0 role: **configure**, **copy**, **dir**, **enable**, **ping**, **show**, **ssh**, **telnet**, **terminal**, **traceroute**, **end**, and **exit**.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **role name priv-***n*<br><br>**Example:**<br>`switch(config)# role name priv-5`<br>`switch(config-role)#` | Enables or disables a privilege role and enters role configuration mode. The *n* argument specifies the privilege level and is a number between 0 and 13. |
| **Step 3** | **rule** *number* {**deny** \| **permit**} **command** *command-string*<br><br>**Example:**<br>`switch(config-role)# rule 2 permit`<br>`command pwd` | Configures a command rule for users of privilege roles. These rules permit or deny users to execute specific commands. You can configure up to 256 rules for each role. The rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.<br><br>The *command-string* argument can contain spaces.<br><br>**Note**    Repeat this command for as many rules as needed. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config-role)# exit`<br>`switch(config)#` | Exits role configuration mode. |
| **Step 5** | (Optional)  **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

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

**Related Topics**

Creating User Roles and Rules, on page 228

# Manually Monitoring TACACS+ Servers or Groups

You can manually issue a test message to a TACACS+ server or to a server group.

**Before you begin**

Enable TACACS+.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **test aaa server tacacs**+ {*ipv4-address* \| *ipv6-address* \| *hostname*} [**vrf** *vrf-name*] *username password*<br><br>**Example:**<br>switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH | Sends a test message to a TACACS+ server to confirm availability. |
| **Step 2** | **test aaa group** *group-name username password*<br>**Example:**<br>switch# test aaa group TacGroup user2 As3He3CI | Sends a test message to a TACACS+ server group to confirm availability. |

**Related Topics**

Configuring TACACS+ Server Hosts, on page 90
Configuring TACACS+ Server Groups, on page 93

# Disabling TACACS+

You can disable TACACS+.

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

**Procedure**

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

# Monitoring TACACS+ Servers

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

**Before you begin**

Configure TACACS+ servers on the Cisco NX-OS device.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| Step 1 | **show tacacs-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br><br>`switch# show tacacs-server statistics`<br>`10.10.1.1` | Displays the TACACS+ statistics. |

**Related Topics**

# Clearing TACACS+ Server Statistics

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

**Before you begin**

Configure TACACS+ servers on the Cisco NX-OS device.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | (Optional) **show tacacs-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>`switch# show tacacs-server statistics 10.10.1.1` | Displays the TACACS+ server statistics on the Cisco NX-OS device. |
| **Step 2** | **clear tacacs-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>`switch# clear tacacs-server statistics 10.10.1.1` | Clears the TACACS+ server statistics. |

**Related Topics**

Configuring TACACS+ Server Hosts, on page 90

# Verifying the TACACS+ Configuration

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

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

# Configuration Examples for TACACS+

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

```
feature tacacs+
tacacs-server key 7 "ToIkLhPpG"
tacacs-server host 10.10.2.2 key 7 "ShMoMhTl"
aaa group server tacacs+ TacServer
    server 10.10.2.2
```

The following example shows how to configure and use command authorization verification:

```
switch# terminal verify-only
switch# show interface ethernet 7/2 brief
%Success
switch# terminal no verify-only
switch# show interface ethernet 7/2 brief


--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                        Speed     Port
Interface                                                                        Ch #
--------------------------------------------------------------------------------
Eth7/2        1      eth  access down    SFP not inserted              auto(D)   --
```

The following example shows how to permit all users with roles priv-5 and above to execute the **pwd** command:

```
switch# configure terminal
switch(config)# role name priv-5
switch(config-role)# rule 1 permit command pwd
```

The following example shows how to deny the **show running-config** command to all users with roles below priv-5. First, you must remove the permission to execute this command from the priv-0 role; then you must permit the command at role priv-5 so that users with roles priv-5 and above have permission to run the command.

```
switch# configure terminal
switch(config)# role name priv-0
switch(config-role)# rule 2 deny command show running-config
switch(config-role)# exit
switch(config)# role name priv-5
switch(config-role)# rule 3 permit command show running-config
switch(config-role)# exit
```

# Where to Go Next

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

# Additional References for TACACS+

This section includes additional information related to implementing TACACS+.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco NX-OS 9000 Series NX-OS Unicast Routing Configuration Guide* |

**Standards**

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

**MIBs**

| MIBs | MIBs Link |
|---|---|
| MIBs related to TACACS+ | To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

**CHAPTER 7**

# Configuring LDAP

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

# About LDAP

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

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

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

## LDAP Authentication and Authorization

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

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

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

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

# LDAP Operation for User Login

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

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

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

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

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

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

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

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

> **Note**   In LDAP, authorization can occur before authentication.

## LDAP Server Monitoring

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

**Figure 4: LDAP Server States**



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

## Vendor-Specific Attributes for LDAP

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

## Cisco VSA Format for LDAP

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

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

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

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

The Cisco NX-OS software supports the following attribute:

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

## Virtualization Support for LDAP

The Cisco NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.*

## Prerequisites for LDAP

LDAP has the following prerequisites:

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

## Guidelines and Limitations for LDAP

LDAP has the following guidelines and limitations:

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

- Cisco NX-OS supports only LDAP version 3.

- Cisco NX-OS supports only these LDAP servers:

    - OpenLDAP

    - Microsoft Active Directory

- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) version 1.2.

- Beginning with Cisco NX-OS Release 10.4(3)F, LDAP over Secure Sockets Layer (SSL) supports TLS version 1.3 and 1.2 on Cisco Nexus switches. TLS v1.1 is deprecated.

- For LDAP over SSL, the LDAP client configuration must include the hostname as a subject in the LDAP server certificate.

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

• Beginning with Cisco NX-OS Release 10.3(1)F, LDAP is supported on the Cisco Nexus 9808 switches.

  • Beginning with Cisco NX-OS Release 10.4(1)F, LDAP is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

• Beginning with Cisco NX-OS Release 10.4(1)F, LDAP is supported on the Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

• Beginning from Cisco NX-OS Release 10.4(2)F, the configuration replace feature is supported for LDAP on Cisco NX-OS devices.

# Default Settings for LDAP

This table lists the default settings for LDAP parameters.

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

# Configuring LDAP

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

# LDAP Server Configuration Process

You can configure LDAP servers by following this configuration process.

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

**Related Topics**

# Enabling or Disabling LDAP

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

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Required: [**no**] **feature ldap**<br><br>**Example:**<br><br>`switch(config)# feature ldap` | Enables LDAP. Use the **no** form of this command to disable LDAP.<br><br>**Note** When you disable LDAP, all related configurations are automatically discarded. |
| **Step 3** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring LDAP Server Hosts

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

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

**Before you begin**

Enable LDAP.

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

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

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ldap-server host** {*ipv4-address* \| *ipv6-address* \| *host-name*} [**enable-ssl**] [**referral-disable**]<br><br>**Example:**<br><br>`switch(config)# ldap-server host`<br>`10.10.2.2 enable-ssl` | Specifies the IPv4 or IPv6 address or hostname for an LDAP server.<br><br>The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish an SSL session prior to sending the bind or search request.<br><br>The **referral-disable** keyword disables the unwanted referral links. |
| **Step 3** | (Optional) **show ldap-server**<br><br>**Example:**<br><br>`switch(config)# show ldap-server` | Displays the LDAP server configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

LDAP Server Configuration Process, on page 119

# Configuring the RootDN for an LDAP Server

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

**Before you begin**

Enable LDAP.

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ldap-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] \| **timeout** *seconds*]]<br><br>**Example:**<br><br>`switch(config)# ldap-server host`<br>`10.10.1.1 rootDN`<br>`cn=manager,dc=acme,dc=com password`<br>`Ur2Gd2BH timeout 60` | Specifies the rootDN for the LDAP server database and the bind password for the root.<br><br>Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. |
| **Step 3** | (Optional) **show ldap-server**<br><br>**Example:**<br><br>`switch(config)# show ldap-server` | Displays the LDAP server configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring LDAP Server Groups

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

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

**Before you begin**

Enable LDAP.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** **Example:** ``` switch# configure terminal switch(config)# ``` | Enters global configuration mode. |
| **Step 2** | [**no**] **aaa group server ldap** *group-name* **Example:** ``` switch(config)# aaa group server ldap LDAPServer1 switch(config-ldap)# ``` | Creates an LDAP server group and enters the LDAP server group configuration mode for that group. |
| **Step 3** | [**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** ``` switch(config-ldap)# server 10.10.2.2 ``` | Configures the LDAP server as a member of the LDAP server group. If the specified LDAP server is not found, configure it using the **ldap-server host** command and retry this command. |
| **Step 4** | (Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]} **Example:** ``` switch(config-ldap)# authentication compare password-attribute TyuL8r ``` | Performs LDAP authentication using the bind or compare method. The default LDAP authentication method is the bind method using first search and then bind. |
| **Step 5** | (Optional) [**no**] **enable user-server-group** **Example:** ``` switch(config-ldap)# enable user-server-group ``` | Enables group validation. The group name should be configured in the LDAP server. Users can login through public-key authentication only if the username is listed as |

| | Command or Action | Purpose |
|---|---|---|
| | | a member of this configured group in the LDAP server. |
| **Step 6** | (Optional) [**no**] **enable Cert-DN-match**<br><br>**Example:**<br>`switch(config-ldap)# enable Cert-DN-match` | Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login. |
| **Step 7** | (Optional) [**no**] **use-vrf** *vrf-name*<br><br>**Example:**<br>`switch(config-ldap)# use-vrf vrf1` | Specifies the VRF to use to contact the servers in the server group. |
| **Step 8** | **exit**<br><br>**Example:**<br>`switch(config-ldap)# exit`<br>`switch(config)#` | Exits LDAP server group configuration mode. |
| **Step 9** | (Optional) **show ldap-server groups**<br><br>**Example:**<br>`switch(config)# show ldap-server groups` | Displays the LDAP server group configuration. |
| **Step 10** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring the Global LDAP Timeout Interval

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

**Before you begin**

Enable LDAP.

**Procedure**

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

**Related Topics**

# Configuring the Timeout Interval for an LDAP Server

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

**Before you begin**

Enable LDAP.

**Procedure**

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

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

**Related Topics**

# Configuring TCP Ports

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

**Before you begin**

Enable LDAP.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ldap-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **port** *tcp-port* [**timeout** *seconds*]<br><br>**Example:**<br>`switch(config)# ldap-server host`<br>`10.10.1.1 port 200 timeout 5` | Specifies the TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535.<br><br>Optionally specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. |

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

**Related Topics**

# Configuring LDAP Search Maps

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

**Before you begin**

Enable LDAP.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **ldap search-map** *map-name*<br><br>**Example:**<br><br>`switch(config)# ldap search-map map1`<br>`switch(config-ldap-search-map)#` | Configures an LDAP search map. |
| **Step 3** | (Optional) [**userprofile** \| **trustedCert** \| **CRLLookup** \| **user-certdn-match** \| **user-pubkey-match** \| **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name*<br><br>**Example:** | Configures the attribute name, search filter, and base-DN for the user profile, trusted certificate, CRL, certificate DN match, public key match, or user-switchgroup lookup search operation. These values are used to send a search query to the LDAP server. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config-ldap-search-map)# userprofile attribute-name att-name search-filter (&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com | The *attribute-name* argument is the name of the attribute in the LDAP server that contains the Nexus role definition. |
| Step 4 | (Optional) **exit** **Example:** switch(config-ldap-search-map)# exit switch(config)# | Exits LDAP search map configuration mode. |
| Step 5 | (Optional) **show ldap-search-map** **Example:** switch(config)# show ldap-search-map | Displays the configured LDAP search maps. |
| Step 6 | (Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Periodic LDAP Server Monitoring

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

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

**Before you begin**

Enable LDAP.

**Procedure**

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

| | Command or Action | Purpose |
|---|---|---|
| | `switch# configure terminal`<br>`switch(config)#` | |
| **Step 2** | Required: [**no**] **ldap-server host** {*ipv4-address* \| *ipv6-address* \| *hostname*} **test rootDN** *root-name* [**idle-time** *minutes* \| **password** *password* [**idle-time** *minutes*] \| **username** *name* [**password** *password* [**idle-time** *minutes*]]]<br><br>**Example:**<br>`switch(config)# ldap-server host`<br>`10.10.1.1 test rootDN root1 username`<br>`user1 password Ur2Gd2BH idle-time 3` | Specifies the parameters for server monitoring. The default username is test, and the default password is Cisco. The default value for the idle timer is 60 minutes, and the valid range is from 1 to 1440 minutes.<br><br>**Note**   We recommend that the user not be an existing user in the LDAP server database. |
| **Step 3** | [**no**] **ldap-server deadtime** *minutes*<br><br>**Example:**<br>`switch(config)# ldap-server deadtime 5` | Specifies the number of minutes before the Cisco NX-OS device checks an LDAP server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 60 minutes. |
| **Step 4** | (Optional) **show ldap-server**<br><br>**Example:**<br>`switch(config)# show ldap-server` | Displays the LDAP server configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

LDAP Server Configuration Process, on page 119

Enabling or Disabling LDAP, on page 120

Configuring LDAP Server Hosts, on page 121

# Configuring the LDAP Dead-Time Interval

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

---

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

---

**Before you begin**

Enable LDAP.

**Procedure**

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

**Related Topics**

# Configuring AAA Authorization on LDAP Servers

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

**Before you begin**

Enable LDAP.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **aaa authorization** {**ssh-certificate** \|<br>**ssh-publickey**} **default** {**group** *group-list* \|<br>**local**}<br><br>**Example:**<br><br>switch(config)# aaa authorization<br>ssh-certificate default group LDAPServer1<br>LDAPServer2 | Configures the default AAA authorization method for the LDAP servers.<br><br>The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the **ssh-publickey** keyword configures LDAP or local authorization with the SSH public key. The default authorization |

| | Command or Action | Purpose |
|---|---|---|
| | | is local authorization, which is the list of authorized commands for the user's assigned role. |
| | | The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The **local** method uses the local database for authorization. |
| Step 3 | (Optional) **show aaa authorization** [**all**]<br><br>**Example:**<br>`switch(config)# show aaa authorization` | Displays the AAA authorization configuration. The **all** keyword displays the default values. |
| Step 4 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring LDAP SSH Public Key Authorization

The AAA authorization is performed through LDAP servers with the public key of the user which is saved in the user entry of the LDAP server.

Before configuring LDAP SSH public key authorization, ensure that the following are taken care of:

- Save the public key of the user as a user attribute in the LDAP server.

- Sign-in using the private key from the SSH client.

**Note** The private key that is presented during SSH sign-in is verified with the public key which is saved in the LDAP server.

The following example shows the sample LDAP client configuration.

In the following example, the public key of the user is saved in the LDAP server under the attribute mentioned in **user-pubkey-match** configuration, ie, **sshPublicKeys** attribute in the below case:

```
ldap-server host fully qualified domain name.com rootDN
"CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com"  password 7 password1
ldap search-map Map1
  userprofile attribute-name "description" search-filter "(cn=$userid)" base-DN
"DC=PI-Sec-DT,DC=com"
  user-pubkey-match attribute-name "sshPublicKeys" search-filter "(cn=$userid)" base-DN
"DC=PI-Sec-DT,DC=com"
aaa group server ldap ldap1
```

```
     server fully qualified domain name.com
     use-vrf management
     ldap-search-map Map1

aaa authorization ssh-publickey default group ldap1
```

In the following example, the SSH client private key of the user is used to sign in to the switch management IP address:

```
ssh ldapuser@10.0.0.1 -i ldap_pub_key_test
```

# Configuring LDAP SSH Certificate Authorization

AAA authorization is performed through an LDAP server with a certificate and the DN of the certificate which is saved in the user attribute of the LDAP server.

During LDAP SSH certificate authorization, following things are taken care of:

- Validation of the user certificate presented through the SSH client using the CA certificate installed in the switch.

- As the **enable cert-dn-match** configuration is enabled by default, the cert-DN-match with the DN stored in the LDAP server to validate the certificate is taken care automatically.

The following example shows the sample LDAP client configurations.

- The following example shows how to save the certificate DN in an LDAP server under any specific attribute that is mentioned in the **user-certdn-match** configuration.

  The format is "x509v3-sign-rsa DN /DC=com, DC=PI-Sec-DT, CN=Users, CN=username1".

  ```
  ldap-server host fully qualified domain name.com rootDN
  "CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com"  password 7 password1
  ldap search-map Map24
    userprofile attribute-name "description" search-filter "(cn=$userid)" base-DN
  "DC=PI-Sec-DT,DC=com"
    user-certdn-match attribute-name <attribute> search-filter "(cn=$userid)" base-DN
  "DC=PI-Sec-DT,DC=com"
  aaa group server ldap ldap24
    server fully qualified domain name.com
    enable Cert-DN-match
    use-vrf management
    ldap-search-map Map24

  aaa authorization ssh-certificate default group ldap24
  ```

- The following show command shows the details of the rootCA certificate installed on the box:

  ```
  switch# show crypto ca certificates
  Trustpoint: ldap
  CA certificate 0:
  subject=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
  issuer=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
  serial=82EE7603BF7E74A9
  notBefore=May 29 07:12:30 2023 GMT
  notAfter=May 26 07:12:30 2033 GMT
  SHA1 Fingerprint=D5:AE:75:8E:A1:4F:79:1E:80:3E:5E:67:C5:42:44:10:13:C6:F7:1D
  purposes: sslserver sslclient

  n7700-DE#
  ```

- The following example shows how user sign-in is performed from the SSH client:

• In the SSH client, the input certificate contains both private key and user certificate concatenated in a single file '<user>.crt'.

• The rootCA.crt is the rootCA certificate file.

• The IP Address is the switch management IP address.

```
ssh username1@10.0.0.1 -i username1.crt -vvv -oCACertificateFile=rootCA.crt
```

# Monitoring LDAP Servers

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

**Before you begin**

Configure LDAP servers on the Cisco NX-OS device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **show ldap-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>`switch# show ldap-server statistics`<br>`10.10.1.1` | Displays the LDAP server statistics. |

**Related Topics**

# Clearing LDAP Server Statistics

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

**Before you begin**

Configure LDAP servers on the Cisco NX-OS device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | (Optional)  **show ldap-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:** | Displays the LDAP server statistics. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# show ldap-server statistics 10.10.1.1` | |
| Step 2 | **clear ldap-server statistics** {*hostname* \| *ipv4-address* \| *ipv6-address*}<br><br>**Example:**<br>`switch# clear ldap-server statistics 10.10.1.1` | Clears the LDAP server statistics. |

**Related Topics**

# Verifying the LDAP Configuration

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

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

# Configuration Examples for LDAP

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

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

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

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

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

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

The following example shows how you can validate the authentication:

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

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

# Where to Go Next

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

# Additional References for LDAP

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

### Standards

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

### MIBs

| MIBs | MIBs Link |
|---|---|
| MIBs related to LDAP | To locate and download the supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

**CHAPTER 8**

# Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Cisco NX-OS devices.

This chapter includes the following sections:

# About SSH and Telnet

This section includes information about SSH and Telnet.

## SSH Server

You can use the 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 user authentication mechanisms supported for SSH are RADIUS, TACACS+, LDAP, and the use of locally stored usernames and passwords.

## SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS device or to any other device that runs the 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 NX-OS software works with publicly and commercially available SSH servers.

# SSH Server Keys

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

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

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

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

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

SSH supports the following public key formats:

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

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

# SSH Authentication Using Digital Certificates

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

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs configured and if not revoked or expired.

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

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

## Telnet Server

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

The Telnet server is disabled by default on the Cisco NX-OS device.

# Prerequisites for SSH and Telnet

Make sure that you have configured IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

# Guidelines and Limitations for SSH and Telnet

SSH and Telnet have the following configuration guidelines and limitations:

- The Cisco NX-OS software supports only SSH version 2 (SSHv2).

- Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS.

- When you use the **no feature ssh feature** command, port 22 is not disabled . Port 22 is always open and a deny rule is pushed to deny all incoming external connections.

- Due to a Poodle vulnerability, SSLv3 is no longer supported.

- IPSG is not supported on the following:

    - The last six 40-Gb physical ports on the Cisco Nexus 9372PX, 9372TX, and 9332PQ switches

    - All 40G physical ports on the Cisco Nexus 9396PX, 9396TX, and 93128TX switches

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

- The SFTP server feature does not support the regular SFTP **chown** and **chgrp** commands.

- When the SFTP server is enabled, only the admin user can use SFTP to access the device.

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

- SSH timeout period must be longer than the time of the tac-pac generation time. Otherwise, the VSH log might show %VSHD-2-VSHD_SYSLOG_EOL_ERR error. Ideally, set to 0 (infinity) before collecting tac-pac or showtech.

**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.

- Beginning with Cisco NX-OS Release 10.2(2)F, a new desynchronization CLI is introduced to provide you an option to disable the user synchronization between the SNMP and the security components. For more information, refer to the *Configuring SNMP* chapter in the *System Management Configuration Guide*.

  For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- When the desynchronization CLI is enabled, remote users will not be synced to SNMP database.

- The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a) will not have a corresponding SNMPv3 profile when the desynchronization CLI is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization CLI along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

# Default Settings for SSH and Telnet

This table lists the default settings for SSH and Telnet parameters.

**Table 10: Default SSH and Telnet Parameters**

| Parameters | Default |
|---|---|
| SSH server | Enabled |
| SSH server key | RSA key generated with 1024 bits |
| RSA key bits for generation | 1024 |
| Telnet server | Disabled |
| Telnet port number | 23 |
| Maximum number of SSH login attempts | 3 |
| SCP server | Disabled |
| SFTP server | Disabled |

# Configuring SSH

This section describes how to configure 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** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **no feature ssh**<br><br>**Example:**<br><br>`switch(config)# no feature ssh` | Disables SSH. |
| **Step 3** | **ssh key** {**dsa** [**force**] \| **rsa** [*bits*[**force**]] \| **ecdsa** [*bits* [ **force**]]}<br><br>**Example:**<br><br>`switch(config)# ssh key rsa 2048` | Generates the SSH server key.<br><br>The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 4096. The default value is 1024.<br><br>You cannot specify the size of the DSA key. It is always set to 1024 bits.<br><br>Use the **force** keyword to replace an existing key.<br><br>**Note**   If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all |
| **Step 4** | **ssh rekey max-data** *max-data* **max-time** *max-time*i<br><br>**Example:**<br><br>`switch(config)# ssh rekey max-data 1K`<br>`max-time 1M` | Configures the rekey parameters. |
| **Step 5** | **feature ssh**<br><br>**Example:**<br><br>`switch(config)# feature ssh` | Enables SSH. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 7** | (Optional) **show ssh key** [**dsa** \| **rsa** \| **ecdsa**] [**md5** ]<br><br>**Example:**<br>`switch# show ssh key` | Displays the SSH server keys.<br><br>This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility. |
| **Step 8** | **show run security all** | |
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# 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 these formats:

- OpenSSH format
- IETF SECSH format

## Specifying the SSH Public Keys in IETF SECSH Format

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

### Before you begin

Generate an SSH public key in IETF SCHSH format.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **copy** *server-file* **bootflash:***filename*<br><br>**Example:**<br>`switch# copy`<br>`tftp://10.10.1.1/secsh_file.pub`<br>`bootflash:secsh_file.pub` | Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 3** | **username** *username* **sshkey file**<br>**bootflash:***filename*<br><br>**Example:**<br>`switch(config)# username User1 sshkey`<br>`file bootflash:secsh_file.pub` | Configures the SSH public key in IETF SECSH format. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 5** | (Optional) **show user-account**<br><br>**Example:**<br>`switch# show user-account` | Displays the user account configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

## Specifying the SSH Public Keys in OpenSSH Format

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

### Before you begin

Generate an SSH public key in OpenSSH format.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **username** *username* **sshkey** *ssh-key*<br><br>**Example:**<br>`switch(config)# username User1 sshkey`<br>`ssh-rsa`<br>`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19cF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEP`<br>`hOByhsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRe5Y/G+lNUQ8g9iqG30c6k6+` | Configures the SSH public key in OpenSSH format. |

| | Command or Action | Purpose |
|---|---|---|
| | XVn+NjnI1B7ihvpVh7dLdbMOXwOnXHYshXrrSiH3UD/vKyziEh5S4Tplx8= | |
| Step 3 | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| Step 4 | (Optional) **show user-account**<br><br>**Example:**<br><br>`switch# show user-account` | Displays the user account configuration. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring a Maximum Number of SSH Login Attempts

You can configure the maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

**Note** The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **ssh login-attempts** *number*<br><br>**Example:**<br><br>`switch(config)# ssh login-attempts 5` | Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10. |

| | Command or Action | Purpose |
|---|---|---|
| | **Note** The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3. | |
| **Step 3** | (Optional) **show running-config security all**<br><br>**Example:**<br>`switch(config)# show running-config security all` | Displays the configured maximum number of SSH login attempts. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Starting SSH Sessions

You can start SSH sessions using IPv4 or IPv6 to connect to remote devices from the Cisco NX-OS device.

**Before you begin**

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

Enable the SSH server on the remote device.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **ssh** [*username@*]{*ipv4-address* \| *hostname*} [**vrf** *vrf-name*]<br><br>**Example:**<br>`switch# ssh 10.10.1.1` | Creates an SSH IPv4 session to a remote device using IPv4. The default VRF is the default VRF. |
| **Step 2** | **ssh6** [*username@*]{*ipv6-address* \| *hostname*} [**vrf** *vrf-name*]<br><br>**Example:**<br>`switch# ssh6 HostA` | Creates an SSH IPv6 session to a remote device using IPv6. |

# Starting SSH Sessions from Boot Mode

You can start SSH sessions from the boot mode of the Cisco NX-OS device to connect to remote devices.

**Before you begin**

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

Enable the SSH server on the remote device.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **ssh** [*username@*]*hostname*<br><br>**Example:**<br>`switch(boot)# ssh user1@10.10.1.1` | Creates an SSH session to a remote device from the boot mode of the Cisco NX-OS device. The default VRF is always used. |
| Step 2 | **exit**<br><br>**Example:**<br>`switch(boot)# exit` | Exits boot mode. |
| Step 3 | **copy scp:**//[*username@*]*hostname*/*filepath directory*<br><br>**Example:**<br>`switch# copy scp://user1@10.10.1.1/users abc` | Copies a file from the Cisco NX-OS device to a remote device using the Secure Copy Protocol (SCP). The default VRF is always used. |

# Configuring SSH Passwordless File Copy

You can copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password. To do so, you must create an RSA or DSA identity that consists of public and private keys for authentication with SSH.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | [**no**] **username** *username* **keypair generate** {**rsa** [*bits* [**force**]] \| **dsa** [**force**]}<br><br>**Example:**<br>`switch(config)# username user1 keypair`<br>`generate rsa 2048 force` | Generates the SSH public and private keys and stores them in the home directory ($HOME/.ssh) of the Cisco NX-OS device for the specified user. The Cisco NX-OS device uses the keys to communicate with the SSH server on the remote machine.<br><br>The *bits* argument is the number of bits used to generate the key. The range is from 768 to 4096. The default value is 1024.<br><br>Use the **force** keyword to replace an existing key. The SSH keys are not generated if the **force** keyword is omitted and SSH keys are already present. |

| | | **Command or Action** | **Purpose** |
|---|---|---|---|
| **Step 3** | | (Optional) **show username** *username* **keypair** | Displays the public key for the specified user. |
| | | **Example:** | **Note** For security reasons, this command does not show the private key. |
| | | switch(config)# show username user1 keypair | |
| **Step 4** | | Required: **username** *username* **keypair export** {**bootflash:***filename* \| **volatile:***filename*} {**rsa** \| **dsa**} [**force**] | Exports the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash or volatile directory. |
| | | **Example:** | Use the **force** keyword to replace an existing key. The SSH keys are not exported if the **force** keyword is omitted and SSH keys are already present. |
| | | switch(config)# username user1 keypair export bootflash:key_rsa rsa | To export the generated key pair, you are prompted to enter a passphrase that encrypts the private key. The private key is exported as the file that you specify, and the public key is exported with the same filename followed by a .pub extension. You can now copy this key pair to any Cisco NX-OS device and use SCP or SFTP to copy the public key file (*.pub) to the home directory of the server. |
| | | | **Note** For security reasons, this command can be executed only from global configuration mode. |
| **Step 5** | | Required: **username** *username* **keypair import** {**bootflash:***filename* \| **volatile:***filename*} {**rsa** \| **dsa**} [**force**] | Imports the exported public and private keys from the specified bootflash or volatile directory to the home directory of the Cisco NX-OS device. |
| | | **Example:** | Use the **force** keyword to replace an existing key. The SSH keys are not imported if the **force** keyword is omitted and SSH keys are already present. |
| | | switch(config)# username user1 keypair import bootflash:key_rsa rsa | To import the generated key pair, you are prompted to enter a passphrase that decrypts the private key. The private key is imported as the file that you specify, and the public key is imported with the same filename followed by a .pub extension. |
| | | | **Note** For security reasons, this command can be executed only from global configuration mode. |
| | | | **Note** Only the users whose keys are configured on the server are able to access the server without a password. |

**What to do next**

On the SCP or SFTP server, use the following command to append the public key stored in the *.pub file (for example, key_rsa.pub) to the authorized_keys file:

**$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys**

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

# Configuring SCP and SFTP Servers

You can configure an SCP or SFTP server on the Cisco NX-OS device in order to copy files to and from a remote device. After you enable the SCP or SFTP server, you can execute an SCP or SFTP command on the remote device to copy the files to or from the Cisco NX-OS device.

**Note**   The arcfour and blowfish cipher options are not supported for the SCP server.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **feature scp-server**<br><br>**Example:**<br><br>`switch(config)# feature scp-server` | Enables or disables the SCP server on the Cisco NX-OS device. |
| **Step 3** | Required: [**no**] **feature sftp-server**<br><br>**Example:**<br><br>`switch(config)# feature sftp-server` | Enables or disables the SFTP server on the Cisco NX-OS device. |
| **Step 4** | Required: **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 5** | (Optional) **show running-config security**<br><br>**Example:**<br><br>`switch# show running-config security` | Displays the configuration status of the SCP and SFTP servers. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring X.509v3 Certificate-Based SSH Authentication

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

### Before you begin

Enable the SSH server on the remote device.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **username** *user-id* [**password** [**0** \| **5**] *password*]<br><br>**Example:**<br><br>`switch(config)# username jsmith password`<br>` 4Ty18Rnt` | Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters for both local and remote users. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (\_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.<br><br>Usernames must begin with an alphanumeric character.<br><br>The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text).<br><br>**Note**    If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device.<br><br>**Note**    If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.<br><br>**Note**    When the desynchronization CLI is enabled, if you create a user account, the corresponding SNMP user will not be created. |
| **Step 3** | **username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** \| **rsa**}<br><br>**Example:** | Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# username jsmith`<br>`ssh-cert-dn "/O = ABCcompany, OU = ABC1,`<br>`emailAddress = jsmith@ABCcompany.com,`<br>`L = Metropolis, ST = New York, C = US,`<br>`CN = jsmith" rsa` | can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively. |
| Step 4 | [**no**] **crypto ca trustpoint** *trustpoint*<br><br>**Example:**<br>`switch(config)# crypto ca trustpoint`<br>`winca`<br>`switch(config-trustpoint)#` | Configures a trustpoint.<br><br>**Note**    Before you delete a trustpoint using the **no** form of this command, you must first delete the CRL and CA certificate, using the **delete crl** and **delete ca-certificate** commands. |
| Step 5 | **crypto ca authenticate** *trustpoint*<br><br>**Example:**<br>`switch(config-trustpoint)# crypto ca`<br>`authenticate winca` | Configures a CA certificate for the trustpoint.<br><br>**Note**    To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode. |
| Step 6 | (Optional) **crypto ca crl request** *trustpoint* **bootflash:***static-crl*.**crl**<br><br>**Example:**<br>`switch(config-trustpoint)# crypto ca`<br>`crl request winca bootflash:crllist.crl` | This command is optional but highly recommended. Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA).<br><br>**Note**    Static CRL is the only supported revocation check method.<br><br>**Note**    To delete the CRL, enter the **delete crl** command. |
| Step 7 | (Optional) **show crypto ca certificates**<br><br>**Example:**<br>`switch(config-trustpoint)# show crypto`<br>`ca certificates` | Displays the configured certificate chain and associated trustpoint. |
| Step 8 | (Optional) **show crypto ca crl** *trustpoint*<br><br>**Example:**<br>`switch(config-trustpoint)# show crypto`<br>`ca crl winca` | Displays the contents of the CRL list of the specified trustpoint. |
| Step 9 | (Optional) **show user-account**<br><br>**Example:**<br>`switch(config-trustpoint)# show`<br>`user-account` | Displays configured user account details. |

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

# Configuring SSH-Cert-Authorization on TACACS Servers

Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS. This feature can be enabled using **aaa authorization ssh-certificate default group** *tac-group-name* command. For more information on configuration, see Configuring AAA SSH-Cert-Authorization on TACACS Servers, on page 39.

# Customizing SSH Cryptographic Algorithms

Cisco Nexus 9000 switches support strong algorithms by default. You can choose to remain with the default mode that enables only strong algorithms as defined by Cisco Product Security Baseline (PSB) or allow all supported algorithms. Note that these algorithms are applicable to the incoming server connections. You can also configure support for SSH key exchange algorithms, message authentication codes (MACs), key types, and ciphers.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal` | Enters the global configuration mode. |
| **Step 2** | (Optional) **ssh kexalgos** [**all** \| **key-exchangealgorithm-name**]<br><br>**Example:**<br>`switch(config)# ssh kexalgos ecdhsha2-nistp384` | Use the **all** keyword to enable all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys.<br><br>Supported KexAlgorithmns are:<br><br>    • curve25519-sha256<br><br>    • diffie-hellman-group-exchange-sha256<br><br>    • diffie-hellman-group1-sha1 |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note**    This algorithm isn't supported from Cisco NX-OS Release 9.3(5). Upgrade your SSH client. |
| | | • diffie-hellman-group14-sha1 |
| | | • diffie-hellman-group1-sha1 |
| | | • ecdh-sha2-nistp256 |
| | | • ecdh-sha2-nistp521 |
| | | To enable only the **ecdh-sha2-nistp384** KexAlgorithm, use the **ecdh-sha2-nistp384** keyword. |
| | | **Note**    Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported KexAlgorithmns. From this release, keyword **ecdh-sha2-nistp384** is deprecated. |
| **Step 3** | (Optional) **ssh macs**  [**all** \| **mac-name**]<br><br>**Example:**<br>`switch(config)# ssh macs`<br>`hmacsha2-256-etm@openssh.com` | Enables all supported MACs which are the message authentication codes used to detect traffic modification.<br><br>Supported MACs are:<br><br>• hmac-sha1<br><br>• hmac-sha2-256<br><br>• hmac-sha2-512<br><br>**Note**    Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported MACs. |
| **Step 4** | (Optional) **ssh ciphers** [ **all**  \| **cipher-name** ]<br><br>**Example:**<br>`switch(config)# ssh ciphers aes192-ctr` | Use the **all** keyword to enable all supported ciphers to encrypt the connection.<br><br>Supported ciphers are:<br><br>• aes128-cbc<br><br>• aes192-cbc<br><br>• aes256-cbc<br><br>• aes128-ctr<br><br>• aes192-ctr<br><br>• aes256-ctr<br><br>• aes128-gcm@openssh.com |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported ciphers. From this release, keyword **aes256-gcm** is deprecated. |
| **Step 5** | (Optional) **ssh keytypes [all \| keytype-string]**<br><br>**Example:**<br>`switch(config)# ssh keytypes`<br>`ecdsa-sha2-nistp256` | Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client.<br><br>Supported key types are:<br><br>  • ecdsa-sha2-nistp256<br><br>  • ecdsa-sha2-nistp384<br><br>  • ecdsa-sha2-nistp521<br><br>  • ssh-dss<br><br>  • ssh-rsa<br><br>  **Note** Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported keytypes.<br><br>**Note** To enable rsa, dsa, and ecdsa key types, you need to generate corresponding SSH host keys.<br><br>**Example configuration:**<br>`switch(config)# ssh key rsa 2048`<br>`switch(config)# ssh key dsa`<br>`switch(config)# ssh key ecdsa 256` |

**Example**

Users can check the supported algorithms using **show ssh [ciphers \| macs \| keytypes \| kexalogs \| version]** commands.

```
show ssh ciphers
Cipher                                         Status     FIPS
----------------------------------------------------------
aes128-ctr                                     permitted   yes
aes192-ctr                                        denied   yes
aes256-ctr                                     permitted   yes
aes128-cbc                                        denied   yes
aes192-cbc                                        denied   yes
aes256-cbc                                        denied   yes
aes256-gcm@openssh.com                         permitted   yes
aes128-gcm@openssh.com                         permitted   yes
```

```
                 chacha20-poly1305@openssh.com                permitted      no



        show ssh macMAC                                          Status    FIPS
        ------------------------------------------------------------
        hmac-sha2-256-etm@openssh.com                 permitted       no
        hmac-sha2-512-etm@openssh.com                 permitted       no
        hmac-sha1-etm@openssh.com                     permitted       no
        hmac-sha2-256                                 permitted       yes
        hmac-sha2-512                                 permitted       yes
        hmac-sha1                                     permitted       yes
        hmac-sha1-96                                   unsupported     no
        hmac-md5                                       unsupported     no
        hmac-md5-96                                    unsupported     no
        umac-64@openssh.com                            unsupported     no
        umac-128@openssh.com                           unsupported     no
        hmac-sha1-96-etm@openssh.com                   unsupported     no
        hmac-md5-etm@openssh.com                       unsupported     no
        umac-64-etm@openssh.com                        unsupported     no
        umac-128-etm@openssh.com                       unsupported     no



        show ssh keytypes Keytype                                Status     FIPS
        ------------------------------------------------------------
        ecdsa-sha2-nistp256-cert-v01@openssh.com   permitted      no <<Currently not suppported>>
        ecdsa-sha2-nistp384-cert-v01@openssh.com   permitted      no <<Currently not suppported>>
        ecdsa-sha2-nistp521-cert-v01@openssh.com   permitted      no <<Currently not suppported>>
        ssh-rsa-cert-v01@openssh.com               permitted      no
        ecdsa-sha2-nistp256                        permitted      yes
        ecdsa-sha2-nistp384                        permitted      yes
        ecdsa-sha2-nistp521                        permitted      no
        rsa-sha2-256                               permitted      no
        ssh-rsa                                    permitted      yes
        ssh-dss                                       denied      no
        ssh-ed25519                                unsupported    no
        ssh-ed25519-cert-v01@openssh.com           unsupported    no
        ssh-dss-cert-v01@openssh.com               unsupported    no



        show ssh kexalgos
        KexAlgorithm                                 Status    FIPS
        ------------------------------------------------------------
        curve25519-sha256                             permitted      no
        curve25519-sha256@libssh.org                  permitted      no
        ecdh-sha2-nistp256                            permitted      yes
        ecdh-sha2-nistp384                            permitted      yes
        ecdh-sha2-nistp521                            permitted      yes
        diffie-hellman-group16-sha512                 permitted      yes
        diffie-hellman-group14-sha1                   permitted      yes
        diffie-hellman-group14-sha256                 permitted      no



        show ssh version
        CiscoSSH 1.9.29, OpenSSH_8.3p1, CiscoSSL 1.1.1t.7.2.500
```

## Algorithms Supported - FIPs Mode Enabled

The list of algorithms supported when the FIPs mode is enabled are as follows:

**Table 11: Algorithms Supported - FIPs Mode Enabled**

| Algorithms | Supported | Unsupported |
|---|---|---|
| ciphers | • aes128-ctr<br>• aes256-ctr<br>• aes256-gcm@openssh.com<br>• aes128-gcm@openssh.com | • aes192-ctr<br>• aes128-cbc<br>• aes192-cbc<br>• aes256-cbc |
| hmac | • hmac-sha2-256<br>• hmac-sha2-512<br>• hmac-sha1 | • hmac-sha2-256-etm@openssh.com<br>• hmac-sha2-512-etm@openssh.com<br>• hmac-sha1-etm@openssh.com |
| kexalgo | • ecdh-sha2-nistp256<br>• ecdh-sha2-nistp384<br>• ecdh-sha2-nistp521<br>• diffie-hellman-group16-sha512<br>• diffie-hellman-group14-sha1<br>• diffie-hellman-group14-sha256 | • curve25519-sha256<br>• curve25519-sha256@libssh.org |
| keytypes | • rsa-sha2-256<br>• ssh-rsa<br>• ecdsa-sha2-nistp256<br>• ecdsa-sha2-nistp384<br>• ecdsa-sha2-nistp521 | • ecdsa-sha2-nistp256-cert-v01@openssh.com<br>• ecdsa-sha2-nistp384-cert-v01@openssh.com<br>• ecdsa-sha2-nistp521-cert-v01@openssh.com |

# Changing the Default SSH Server Port

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 2** | **no feature ssh**<br><br>**Example:**<br><br>`switch(config)# no feature ssh` | Disables SSH. |
| **Step 3** | **show sockets** *local-port-range*<br><br>**Example:**<br><br>`switch(config)# show sockets local port`<br>` range (15001 - 58000)`<br>`switch(config)# local port range (58001`<br>` - 63535) and nat port range (63536 -`<br>`65535)`<br><br>`switch# show sockets local-port-range`<br>`Kstack local port range (15001 - 22002)`<br>`Netstack local port range (22003 – 65535)` | Displays the available port range. |
| **Step 4** | **ssh port** *local-port*<br><br>**Example:**<br><br>`switch(config)# ssh port 58003` | Configures the port.<br><br>**Note**    When you upgrade from an earlier release to Release 9.3(1) or later releases, ensure that features with user-defined SSH port, are within the following range:<br><br>     • For Release 9.3(1) and Release 9.3(2): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 - 63535, and nat port range is from 63536 to 65535<br><br>     • From Release 9.3(3): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 to 60535, and nat port range is from 60536 to 65535 |
| **Step 5** | **feature ssh**<br><br>**Example:**<br><br>`switch(config)# feature ssh` | Enables SSH. |
| **Step 6** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 7** | (Optional) **show running-config security all**<br><br>**Example:**<br><br>`switch# ssh port 58003` | Displays the security configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, or when you start an SSH session from this device to a remote host, you establish a trusted SSH relationship with that server. You can clear the list of trusted SSH servers for your user account.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **clear ssh hosts**<br>**Example:**<br>`switch# clear ssh hosts` | Clears the SSH host sessions and the known host file. |

# Disabling the SSH Server

By default, the SSH server is enabled on the Cisco NX-OS device. You can disable the SSH server to prevent SSH access to the switch.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **no feature ssh**<br>**Example:**<br>`switch(config)# no feature ssh` | Disables SSH. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 4** | (Optional) **show ssh server**<br>**Example:**<br>`switch# show ssh server` | Displays the SSH server configuration. |

| | Command or Action | Purpose |
|---|---|---|
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Deleting SSH Server Keys

You can delete SSH server keys on the Cisco NX-OS device 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 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **no feature ssh**<br><br>**Example:**<br>`switch(config)# no feature ssh` | Disables SSH. |
| Step 3 | **no ssh key** [**dsa** \| **rsa** \| **ecdsa**]<br><br>**Example:**<br>`switch(config)# no ssh key rsa` | Deletes the SSH server key.<br><br>The default is to delete all the SSH keys. |
| Step 4 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| Step 5 | (Optional) **show ssh key**<br><br>**Example:**<br>`switch# show ssh key` | Displays the SSH server key configuration. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Clearing SSH Sessions

You can clear SSH sessions from the Cisco NX-OS device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **show users** <br><br> **Example:** <br><br> `switch# show users` | Displays user session information. |
| **Step 2** | **clear line** *vty-line* <br><br> **Example:** <br><br> `switch(config)# clear line pts/12` | Clears a user SSH session. |

# Configuring Telnet

This section describes how to configure Telnet on the Cisco NX-OS device.

# Enabling the Telnet Server

You can enable the Telnet server on the Cisco NX-OS device. By default, the Telnet server is disabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br><br> **Example:** <br><br> `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **feature telnet** <br><br> **Example:** <br><br> `switch(config)# feature telnet` | Enables the Telnet server. The default is disabled. |
| **Step 3** | **exit** <br><br> **Example:** <br><br> `switch(config)# exit`<br>`switch#` | Exits global configuration mode. |

|  | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **show telnet server**<br><br>**Example:**<br>`switch# show telnet server` | Displays the Telnet server configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS device. You can start Telnet sessions using either IPv4 or IPv6.

### Before you begin

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device.

Enable the Telnet server on the Cisco NX-OS device.

Enable the Telnet server on the remote device.

### Procedure

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **telnet** {*ipv4-address* \| *host-name*} [*port-number*] [**vrf** *vrf-name*]<br><br>**Example:**<br>`switch# telnet 10.10.1.1` | Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF. |
| **Step 2** | **telnet6** {*ipv6-address* \| *host-name*} [*port-number*] [**vrf** *vrf-name*]<br><br>**Example:**<br>`switch# telnet6 2001:0DB8::ABCD:1 vrf`<br>`management` | Starts a Telnet session to a remote device using IPv6. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF. |

### Related Topics

Enabling the Telnet Server, on page 159

# Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco NX-OS device.

### Before you begin

Enable the Telnet server on the Cisco NX-OS device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **show users**<br><br>**Example:**<br>`switch# show users` | Displays user session information. |
| **Step 2** | **clear line** *vty-line*<br><br>**Example:**<br>`switch(config)# clear line pts/12` | Clears a user Telnet session. |

# Verifying the SSH and Telnet Configuration

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

| **Command** | **Purpose** |
|---|---|
| **show ssh key** [**dsa** \| **rsa**] [**md5**] | Displays the SSH server keys.<br><br>For Cisco NX-OS Release 7.0(3)I4(6) and 7.0(3)I6(1) and any later releases, this command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility. |
| **show running-config security** [**all**] | Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts. |
| **show ssh server** | Displays the SSH server configuration. |
| **show telnet server** | Displays the Telnet server configuration. |
| **show username** *username* **keypair** | Displays the public key for the specified user. |
| **show user-account** | Displays configured user account details. |
| **show users** | Displays the users logged into the device. |
| **show crypto ca certificates** | Displays the configured CA certificate and associated trustpoint for X.509v3 certificate-based SSH authentication. |
| **show crypto ca crl** *trustpoint* | Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication. |

# Configuration Example for SSH

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

**Procedure**

**Step 1** Disable the SSH server.

**Example:**
```
switch# configure terminal
switch(config)# no feature ssh
```

**Step 2** Generate an SSH server key.

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

**Step 3** Enable the SSH server.

**Example:**
```
switch(config)# feature ssh
```

**Step 4** Display the SSH server key.

**Example:**
```
switch(config)# show ssh key
could not retrieve dsa key information
**************************************
rsa Keys generated:Tue Mar 14 13:13:47 2017

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDh4+DZboQJbJt10nJhgKBYL5lOlhsFM2oZRi9+JqEU
GA44I9ej+E5NIRZ1x8ohIt6Vx9Et5csO7Pw72rjUwR3UPmuAm79k7I/SyLGEP3WUL7sqbLvNF5GqKXph
oqMT075WUdbGWphorA2g0tTObRrFIQBJVQ0SSBh3oEaaALqYUQ==

bitcount:1024
fingerprint:
SHA256:V6KAeLAiKRRUPBZm1Yq3rl6JW7Eo7vhLi6CXYxnD/+Y
**************************************
**************************************


switch(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2013

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr
+MZm99n2U0ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39
HmXL6VgpRVn1XQFiBwn4na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

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

**Step 5** Specify the SSH public key in OpenSSH format.

**Example:**

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50r
v7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQ
W3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5
4Tplx8=
```

**Step 6** Save the configuration.

**Example:**

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

# Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

**Procedure**

**Step 1** Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

**Example:**

```
switch# configure terminal
switch(config)# username admin keypair generate rsa
generating rsa key(1024 bits)......
generated rsa key
```

**Step 2** Display the public key for the specified user.

**Example:**

```
switch(config)# show username admin keypair

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

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

**Step 3** Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

**Example:**

```
switch(config)# username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
.
.
.
      951    Jul 09 11:13:59 2013  key_rsa
      221    Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

**Step 4** After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

**Example:**

```
switch(config)# username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
switch(config)#
```

**Step 5** On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

**Example:**

**$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys**

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

**Step 6** (Optional) Repeat this procedure for the DSA keys.

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

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

**Note** Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS.

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

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

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

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

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

# Additional References for SSH and Telnet

This section describes additional information related to implementing SSH and Telnet.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

**RFCs**

| RFCs | Title |
|---|---|
| RFC 6187 | *X.509v3 Certificates for Secure Shell Authentication* |

**MIBs**

| MIBs | MIBs Link |
|---|---|
| MIBs related to SSH and Telnet | To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

# Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the Cisco NX-OS device. PKI allows the device to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for Secure Shell (SSH).

This chapter includes the following sections:

## Information About PKI

This section provides information about PKI.

## CAs and Digital Certificates

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

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

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

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

# Trust Model, Trust Points, and Identity CAs

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

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

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

# CA Certificate Hierarchy

For secure services, you typically have multiple trusted CAs. The CAs are usually installed in all the hosts as a bundle. The NX-OS PKI infrastructure does support importing certificate chain. However, with the current CLIs, one chain at a time can be installed. This procedure can be cumbersome when there are several CA chains to be installed. This requires a facility to download CA bundles that could include several intermediate and root CAs.

## Importing CA Bundle

The **crypto CA trustpoint** command binds the CA certificates, CRLs, identity certificates and key pairs to a named label. All files corresponding to each of these entities are stored in the NX-OS certstore directory (/isan/etc/certstore) and tagged with the trustpoint label.

To access the CA certificates, an SSL app only needs to point to the standard NX-OS cert-store and specify that as the CA path during SSL initialization. It does not need to be aware of the trustpoint label under which CAs are installed.

If clients need to bind to an identity certificate, the trustpoint label needs to be used as the binding point.

The import pkcs command is enhanced to install the CA certificates under a trustpoint label. This can be further enhanced to install a CA bundle. The import command structure is modified to add pkcs7 option which is used for providing CA bundle file in pkcs7 format.

Beginning with Cisco NX-OS Release 10.1(1), the pkcs7 file format is supported to unpack the CA bundle and install each CA chain under its own label. The labels are formed by appending an index to the main trustpoint label.

Once installed, there is no logical binding of all CA chains to a bundle.

# Import of the CA Certificate Bundle in PKCS7 Format

To support the import of the ca certificate bundle which consists of multiple independent certificate chains, the option of 'pkcs7' is introduced in the crypto import command.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca import** *<baselabel>* **pksc7** *<uri0>* **force** | There are two input arguments in the command. The source file which is the ca bundle file is given in the *<uri0>*, the input file has to be in pkcs7 format indicating that it is a cabundle file.<br><br>Multiple certificate chains will be extracted out of the cabundle. The command will generate multiple trustpoints with ca certificate chain attached to each one. Import command generates two configurations which are global CA bundle configuration and CA bundle sub-configuration with each trustpoint generated.<br><br>The force option removes the CA bundle and related trustpoint configurations, imports a new CA bundle with the same bundle name, and generates fresh trustpoint configurations related to the cabundle. |
| **Step 3** | **crypto ca cabundle** *<bundle-name>* | The bundle-name is same as baselabel for import case. You can use the **no** form of this command to delete the, CA bundle, trustpoints, and related certificate chains.<br><br>After importing CA bundle under a particular baselabel name and generating all the trustpoints, if a user try to execute the import command again under the same baselabel name, it will throw error saying CA bundle already exists. The user can use force option to modify the existing CA bundle.<br><br>Maximum number of Cabundles supported is 20. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **show crypto ca certificates**<br><br>**Example:**<br>`switch# show crypto ca certificates` | Displays the CA certificates. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# RSA Key Pairs and Identity Certificates

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

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

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

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

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

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

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

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

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

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

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

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

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

## Multiple Trusted CA Support

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

## PKI Enrollment Support

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

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

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

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

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

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

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

## Manual Enrollment Using Cut-and-Paste

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

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

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

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

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

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

# Multiple RSA Key Pair and Identity CA Support

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

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

# Peer Certificate Verification

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

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

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

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

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

# Certificate Revocation Checking

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

## CRL Support

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

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

## NDcPP: OCSP for Syslog

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

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

## Import and Export Support for Certificates and Associated Key Pairs

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

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

# Guidelines and Limitations for PKI

PKI has the following configuration guidelines and limitations:

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

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

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

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

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

- Configuration rollbacks do not support the PKI configuration.

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

- Beginning with Cisco NX-OS Release 10.3(3)F, Elliptic Curve Cyptography (ECC) key pair support is provided to generate and import the certificate on Cisco Nexus switches.

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

# Default Settings for PKI

This table lists the default settings for PKI parameters.

**Table 12: Default PKI Parameters**

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

# Configuring CAs and Digital Certificates

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

## Configuring the Hostname and IP Domain Name

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

⚠

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **hostname** *hostname*<br><br>**Example:**<br><br>`switch(config)# hostname DeviceA` | Configures the hostname of the device. |
| **Step 3** | **ip domain-name** *name* [**use-vrf** *vrf-name*]<br><br>**Example:**<br><br>`DeviceA(config)# ip domain-name`<br>`example.com` | Configures the IP domain name of the device. If you do not specify a VRF name, the command uses the default VRF. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show hosts**<br><br>**Example:**<br><br>`switch# show hosts` | Displays the IP domain name. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Generating an RSA Key Pair

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

Beginning Cisco NX-OS Release 9.3(3), you must explicitly generate RSA key pairs before you associate the Cisco NX-OS device with a trust point CA. Prior to Cisco NX-OS Releases 9.3(3), if unavailable, the RSA key pairs would be auto generated.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]<br><br>**Example:**<br><br>`switch(config)# crypto key generate rsa`<br>` exportable` | Generates an RSA key pair. The maximum number of key pairs on a device is 16.<br><br>The label string is alphanumeric, case sensitive, and has a maximum length of 64 characters. |

| | Command or Action | Purpose |
|---|---|---|
| | | The default label string is the hostname and the FQDN separated by a period character (.). |
| | | Valid modulus values are 512, 768, 1024, 1536, 2048, 3072 and 4096. The default modulus size is 512. |
| | | **Note** The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus. |
| | | By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format. |
| | | **Caution** You cannot change the exportability of a key pair. |
| **Step 3** | **exit** <br><br> **Example:** <br> `switch(config)# exit` <br> `switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show crypto key mypubkey rsa** <br><br> **Example:** <br> `switch# show crypto key mypubkey rsa` | Displays the generated key. |
| **Step 5** | (Optional) **copy running-config startup-config** <br><br> **Example:** <br> `switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Generating an ECC Key Pair

You can generate an ECC key pair to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the ECC key pair before you can obtain a certificate for your device. The ECC keys are stronger compared to RSA keys for a given length.

Beginning Cisco NX-OS Release 10.3(3)F, you can generate an ECC key pair to associate the Cisco NX-OS device with a trust point CA.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto key generate ecc** [**label** *ecc-key-label*] [**exportable**] [**modulus** *size*]<br><br>**Example:**<br><br>`switch(config)# crypto key generate ecc`<br>`exportable modulus 224` | Generates an ECC key pair. The maximum number of key pairs on a device is 16.<br><br>The label string is alphanumeric, case sensitive, and has maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.).<br><br>Valid modulus values are 224, 384, and 521. The default modulus size is 224.<br><br>**Note** The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus.<br><br>By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format.<br><br>**Caution** You cannot change the exportability of a key pair. |
| **Step 3** | **no crypto key generate ecc** [**label** *ecc-key-label*]<br><br>**Example:**<br><br>`switch(config)# no crypto key generate`<br>`ecc label label-name` | Deletes the ECC key. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show crypto key mypubkey ecc**<br><br>**Example:**<br><br>`switch# show crypto key mypubkey ecc` | Displays the generated ECC key. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

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

# Creating a Trust Point CA Association

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

**Before you begin**

Generate the RSA key pair.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca trustpoint** *name*<br><br>**Example:**<br><br>`switch(config)# crypto ca trustpoint admin-ca`<br>`switch(config-trustpoint)#` | Declares a trust point CA that the device should trust and enters trust point configuration mode.<br><br>**Note** The maximum number of trustpoints that can be configured is 50. |
| **Step 3** | **cabundle** *baselabel*<br><br>**Example:**<br><br>`switch(config-trustpoint)# cabundle test` | Groups the trustpoints under a specific CA bundle. The **No** form of this command detaches the trustpoints from the CA bundle. This command associates the trustpoints to an existing CA bundle and it does not configure any new CA bundle. |
| **Step 4** | **enrollment terminal**<br><br>**Example:**<br><br>`switch(config-trustpoint)# enrollment terminal` | Enables manual cut-and-paste certificate enrollment. The default is enabled.<br><br>**Note** The Cisco NX-OS software supports only the manual cut-and-paste method for certificate enrollment. |
| **Step 5** | **rsakeypair** *label*<br><br>**Example:**<br><br>`switch(config-trustpoint)# rsakeypair SwitchA` | Specifies the label of the RSA key pair to associate to this trust point for enrollment.<br><br>**Note** You can specify only one RSA key pair per CA. |
| **Step 6** | **exit**<br><br>**Example:**<br><br>`switch(config-trustpoint)# exit`<br>`switch(config)#` | Exits trust point configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | (Optional) **show crypto ca trustpoints**<br><br>**Example:**<br><br>`switch(config)# show crypto ca trustpoints` | Displays trust point information. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Certificate Mapping Filters

You can configure mapping filters to validate the CA certificates that are used for authentication. The mapping filters are used to match the CA certificate against a username.

Cisco NX-OS supports the following certificate mapping filters:

- %username%—Substitutes the user's login name.

- %hostname%—Substitutes the peer hostname.

**Before you begin**

Configure a cert-store for certificate authentication.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto certificatemap mapname** *map-name*<br><br>**Example:**<br><br>`switch(config)# crypto certificatemap mapname filtermap1` | Creates a new filter map. |
| **Step 3** | **filter** [**subject-name** *subject-name* \| **altname-email** *e-mail-ID* \| **altname-upn** *user-principal-name*]<br><br>**Example:**<br><br>`switch(config-certmap-filter)# filter altname-upn %username%@cisco.com` | Configures one or more certificate mapping filters within the filter map. These certificate field attributes are supported in the filters: The validation passes if the certificate passes all of the filters configured in the map. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | • subject-name—The required subject name in the LDAP distinguished name (DN) string format. For example: |
| | | `filter subject-name CN=%username%` |
| | | or |
| | | `filter subject-name /C=IN/ST=KA/L=BLR/O=CISCO/OU=ABC/CN=%username%` |
| | | • altname-email—The e-mail address that must be present in the certificate as a subject alternative name. For example: |
| | | `filter altname-email %username%@cisco.com` |
| | | • altname-upn—The principal name that must be present in the certificate as a subject alternative name. For example: |
| | | `filter altname-upn %username%@%hostname%` |
| | | The validation passes if the certificate passes all of the filters configured in the map. |
| **Step 4** | **exit** <br><br> **Example:** <br><br> `switch(config-certmap-filter)# exit` <br> `switch(config)#` | Exits certificate mapping filter configuration mode. |
| **Step 5** | (Optional) **crypto cert ssh-authorize** [**default** \| *issuer-CAname*] [**map** *map-name1* [*map-name2*]] <br><br> **Example:** <br><br> `switch(config)# crypto cert ssh-authorize default map filtermap1` | Configures a certificate mapping filter for the Secure Shell (SSH) protocol. You can use the default filter map for SSH authorization or specify the issuer of the CA certificate. If you do not use the default map, you can specify one or two filter maps for authorization. <br><br> If you specify the issuer of the CA certificate, the certificate bound to the user account is validated as successful if it passes one of the configured maps. |
| **Step 6** | (Optional) **show crypto certificatemap** <br><br> **Example:** <br><br> `switch(config)# show crypto certificatemap` | Displays the certificate mapping filters. |
| **Step 7** | (Optional) **show crypto ssh-auth-map** <br><br> **Example:** <br><br> `switch(config)# show crypto ssh-auth-map` | Displays the mapping filters configured for SSH authentication. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Authenticating the CA

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

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

**Before you begin**

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca authenticate** *name* **pemfile** *uri0*<br><br>**Example:**<br><br>`switch(config)# crypto ca authenticate`<br>`admin-ca`<br>`input (cut & paste) CA certificate`<br>`(chain) in PEM format;`<br>`end the input with a line containing only`<br>` END OF INPUT :`<br>`-----BEGIN CERTIFICATE-----`<br>`MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRl1jK0ZejANBgkqhkiG9w0BAQUFADCB` | Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.<br><br>Also validates and attaches the CA chain directly to the specified trust point.<br><br>The maximum number of trust points that you can authenticate to a specific CA is 10. |

| | Command or Action | Purpose |
|---|---|---|
| | kDEgMB4GCSqGSIb3DQEJARYRWllHomRrzZLBjaXNjby5jb20xCzAJBgNVBAYTAklC<br>MRIwEAYDVQQIEwlLYXJuYXRha2ExEeEjAQBgNVBAcTCUJhomdhbG9yZTIEOMAwGA1UE<br>CHMFQ2lzY28xEzARBgNVBAsTCHr5ldHN0o3JhZZUkEjAQBgNVBAMTCUFwXJUiYSED<br>QIAeFw0wNTAlMDMyMjQ2MzdaFw0wNzA1MDMyMjUlMTdaMIGQMSAwHgYJKoZIhvcN<br>AQkBFhFhoWFtZGtlQGNpc2NvLmNvbTEIMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth<br>cm5hdGFrYTIESMBAGA1UEBxMJQnFtZ2Fsb3JlIMQ4wDAYDVQQKEwVDaXNjbzEIMBEG<br>A1UECxMKbmV0c3RvcmFnZIESMBAGA1UEAxMJQXBhcmbhIENBMFwwDQYJKoZIhvcN<br>AQEBBQADSwAwSAIBAMW/7b3+DXJFANBsIHtzluNccNM87ypyzwudSNZXOMpeRXXI<br>OzyBAgiXI2ASFuUOwQliIMBrO/41jf8RxvYKvysCAwEAAaOBvzCBvDAIBgNVHQ8E<br>BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFRQUUyjyRoMbrCNVRU2OyRhQ<br>GgsWbHEwawYDVR0fBGQwYjAuoCygKOYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs<br>L0FwYXJuYSUyMENBLmNybDAwoC6gLIYq2mlsZIovL1xxc3NlLIPA4XENlcmREbnJv<br>bGxcQXBhcnbhJIIwQEuY3JSMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB<br>BQUAA0EAHv6UQ+8nE399Iww+KaGr0gONLJaqNgIh0AFcTOrEyuyt/WYGPzksF9Ea<br>NBG7E0oN66zex0EOEfG1Vs6mXp1//w==<br>-----END CERTIFICATE-----<br>END OF INPUT<br>Fingerprint(s): MD5<br>Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12<br>Do you accept this certificate? [yes/no]:<br> yes | **Note** For subordinate CA authentication, the Cisco NX-OS software requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>switch(config)# exit<br>switch# | Exits configuration mode. |
| **Step 4** | (Optional) **show crypto ca trustpoints**<br><br>**Example:**<br><br>switch# show crypto ca trustpoints | Displays the trust point CA information. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# copy running-config<br>startup-config | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring Certificate Revocation Checking Methods

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

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

**Before you begin**

Authenticate the CA.

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca trustpoint** *name*<br><br>**Example:**<br><br>`switch(config)# crypto ca trustpoint`<br>`admin-ca`<br>`switch(config-trustpoint)#` | Specifies a trust point CA and enters trust point configuration mode. |
| **Step 3** | **revocation-check** {**crl** [**none**] | **none**}<br><br>**Example:**<br><br>`switch(config-trustpoint)#`<br>`revocation-check none` | Configures the certificate revocation checking methods. The default method is **crl**.<br><br>The Cisco NX-OS software uses the certificate revocation methods in the order that you specify. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config-trustpoint)# exit`<br>`switch(config)#` | Exits trust point configuration mode. |
| **Step 5** | (Optional) **show crypto ca trustpoints**<br><br>**Example:**<br><br>`switch(config)# show crypto ca`<br>`trustpoints` | Displays the trust point CA information. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Authenticating the CA, on page 181
Configuring a CRL, on page 189

# Generating Certificate Requests

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

**Before you begin**

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca enroll** *name*<br><br>**Example:**<br><br>`switch(config)# crypto ca enroll admin-ca`<br>`Create the certificate request ..`<br>` Create a challenge password. You will`<br>`need to verbally provide this`<br>` password to the CA Administrator in`<br>`order to revoke your certificate.`<br>` For security reasons your password will`<br>` not be saved in the configuration.`<br>` Please make a note of it.`<br>` Password:nbv123`<br>`The subject name in the certificate will`<br>` be: DeviceA.cisco.com`<br>` Include the switch serial number in the`<br>` subject name? [yes/no]: no`<br>` Include an IP address in the subject`<br>`name [yes/no]: yes`<br>`ip address:172.22.31.162`<br>` The certificate request will be`<br>`displayed...`<br>`-----BEGIN CERTIFICATE REQUEST-----`<br>`MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ`<br>`KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UA2NC7jU1lDVaSMqNIguI2kt8r141KY`<br>`0JC6MamNy4qmk8VeMXZSiLJ4JgTzKWdbbDIKDKTTysnrjuCXGvjb+wj0nEhv/y5lT9y`<br>`P2NUO8omqShrvFZgC7ysN/PyMwKcgzhbVpj+rarg2vHtGJ91XIIq4WoVkSCzXv8S`<br>`VqyH0vEvAgMBAAGgITzAVBgkqhkiG9w0BCQcxCBMGbnu2MTIzMDyGCSqGSIb3DQEJ`<br>`DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ`<br>`KoZIhvcNAQEBBQADgYEAkI60KER6Qo8nj0sDXZMHSfJZh6K6JtDz3GkcB9GlFWgt`<br>`PftrNOWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktEkxiI6U188rHIOjg1Mjja8`<br>`8a23bNDpNeM8rklwAGrWkrVL8NUZEKJxdprjfrgRNIZacJOUS6ZqKCMetckKytUkO=`<br>`-----END CERTIFICATE REQUEST-----` | Generates a certificate request for an authenticated CA.<br><br>**Note** You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config-trustpoint)# exit`<br>`switch(config)#` | Exits trust point configuration mode. |
| **Step 4** | (Optional) **show crypto ca certificates**<br><br>**Example:**<br><br>`switch(config)# show crypto ca`<br>`certificates` | Displays the CA certificates. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Installing Identity Certificates

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

**Before you begin**

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca import** *name* **certificate**<br><br>**Example:**<br><br>`switch(config)# crypto ca import admin-ca`<br>` certificate`<br>`input (cut & paste) certificate in PEM`<br>`format:`<br>`-----BEGIN CERTIFICATE-----`<br>`MIIEADCCA6ogAwIBAgIKCjOOoQAAAAAdDANBgkqhkiG9w0BAQUFADDBkDEgMB4G`<br>`CSqGSIb3DQEJARYRYWIhbmRrzZiBjaXNjby5jb20xCzAJBgNVBAYTAklQMRIwEAYD`<br>`VQQIEwIlXXJuiXRna2ExEjAQBgNVBAcTCUJhomdhbcG9yZTEOMAwGA1UEChMFQ21z`<br>`Y28xEzARBgNVBAsTCtr5ldHN0b3JhZ21uExEjAQBgNVBAMTCUFwXJuiYSBDQIAeFw0w`<br>`NIExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu`<br>`Y21zY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADiBiQKBgQC/GWACdjQx41C`<br>`dQIIWkjKjSICdpLfK5eJ5tNCQujGpzcuKsZPFXjF2JoiyeCYE8ylndWyw5E08rJ47`<br>`glxr42/sI9IRIb/8uoU/cj9jSSfKK56koa7xWAu8rDfz83jMChIM4WlaY/q2g4Go`<br>`x7RifoVO6uRqFZEgs17/Elash9LxIwIIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw`<br>`GYIRVMvhXMtMS5jaXNjby5jb22HBKwNH6IwHQYDVR0OBBYEFKCLi+2ssp9WEfgrR`<br>`bWmlVyc9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjIEMNjskYUBoLFmxxcYGW`<br>`pIGIMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE`<br>`BhMCSU4xEjAQBgNVBAgTCUthcmFhdGGFrYIESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w`<br>`DAYDVQQKEwVDaXNjbzEIMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh`<br>`cmfhIENBghAFYNKJrlQZlE9lEiWMtRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6`<br>`Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFyblElMjBDQS5jcmwwMKAuoCyGKmZpbGU6`<br>`Ly9cXHNzZS0wOExDZXJ0RW5yb2xsXEFwXJuYSUyMENBLmNybDDBigYIKwYBBQ1H`<br>`AQFEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTIPA4LQNLcnRFbnJvbGwvc3N1` | Prompts you to cut and paste the identity certificate for the CA named admin-ca.<br><br>The maximum number of identify certificates that you can configure on a device is 16. |

| | Command or Action | Purpose |
|---|---|---|
| | LTPA4XOFwYXJuYSUyMENBImNycDA9BggrBgEFBQcwAoYXZmlsZTovLl1xcc3N1LTPA4<br>XENlcrRFbnJvbGxcc3N1LTPA4XOFwYXJuYSUyMENBImNycDANBgkqhkiG9w0BAQUF<br>AANBADbGBGsbe7GNLh9xeOIWBNbm24U69ZSuLDcOcUZUUIgrqnIIqVpPyejtsyflw<br>E36cIZu4WsExREqxbTk8ycx7V5o=<br>-----END CERTIFICATE----- | |
| Step 3 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| Step 4 | (Optional) **show crypto ca certificates**<br><br>**Example:**<br>`switch# show crypto ca certificates` | Displays the CA certificates. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Creating a Trust Point CA Association, on page 178

# Ensuring Trust Point Configurations Persist Across Reboots

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

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

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

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

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

**Related Topics**

Exporting Identity Information in PKCS 12 Format, on page 187

# Exporting Identity Information in PKCS 12 Format

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

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

**Before you begin**

Authenticate the CA.

Install an identity certificate.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **crypto ca export** *name* **pkcs12 bootflash:***filename password*<br><br>**Example:**<br><br>`switch(config)# crypto ca export admin-ca`<br>` pkcs12 bootflash:adminid.p12 nbv123` | Exports the identity certificate and associated key pair and CA certificates for a trust point CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | **copy booflash:***filename scheme***://***server*/ [*url*/]*filename*<br><br>**Example:**<br><br>`switch# copy bootflash:adminid.p12`<br>`tftp:adminid.p12` | Copies the PKCS#12 format file to a remote server.<br><br>For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.<br><br>The *server*, *url*, and *filename* arguments are case sensitive. |

**Related Topics**

# Importing Identity Information in PKCS 12 or PKCS 7 Format

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

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

**Before you begin**

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

**Procedure**

|        | **Command or Action** | **Purpose** |
|--------|-----------------------|-------------|
| **Step 1** | **copy** *scheme***://** *server*/[*url* /]*filename* **bootflash:***filename*<br><br>**Example:**<br>`switch# copy tftp:adminid.p12 bootflash:adminid.p12` | Copies the PKCS#12 format file from the remote server.<br><br>For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.<br><br>The *server*, *url*, and *filename* arguments are case sensitive. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 3** | **crypto ca import** *name* **[pksc12 | pkcs7] bootflash:***filename*<br><br>**Example:**<br>`switch(config)# crypto ca import admin-ca`<br>` pkcs12 bootflash:adminid.p12 nbv123` | Imports the identity certificate and associated key pair and CA certificates for trust point CA. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show crypto ca certificates**<br><br>**Example:**<br>`switch# show crypto ca certificates` | Displays the CA certificates. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring a CRL

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

**Before you begin**

Ensure that you have enabled certificate revocation checking.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **copy** *scheme***:**[**//***server*/[*url* /]]*filename* **bootflash:***filename*<br><br>**Example:**<br><br>`switch# copy tftp:adminca.crl`<br>`bootflash:adminca.crl` | Downloads the CRL from a remote server.<br><br>For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.<br><br>The *server*, *url*, and *filename* arguments are case sensitive. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 3** | **crypto ca crl request** *name* **bootflash:***filename*<br><br>**Example:**<br><br>`switch(config)# crypto ca crl request`<br>`admin-ca bootflash:adminca.crl` | Configures or replaces the current CRL with the one specified in the file. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show crypto ca crl** *name*<br><br>**Example:** | Displays the CA CRL information. |

| | Command or Action | Purpose |
|---|---|---|
| | switch# show crypto ca crl admin-ca | |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Deleting Certificates from the CA Configuration

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **crypto ca trustpoint** *name*<br><br>**Example:**<br><br>switch(config)# crypto ca trustpoint admin-ca<br>switch(config-trustpoint)# | Specifies a trust point CA and enters trust point configuration mode. |
| **Step 3** | **delete ca-certificate**<br><br>**Example:**<br><br>switch(config-trustpoint)# delete ca-certificate | Deletes the CA certificate or certificate chain. |
| **Step 4** | **delete certificate** [**force**]<br><br>**Example:**<br><br>switch(config-trustpoint)# delete certificate | Deletes the identity certificate.<br><br>You must use the **force** option if the identity certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only the identity certificate and leave the applications (such as SSH) without a certificate to use. |
| **Step 5** | **exit**<br><br>**Example:** | Exits trust point configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config-trustpoint)# exit<br>switch(config)# | |
| Step 6 | (Optional) **show crypto ca certificates** [*name*]<br><br>**Example:**<br>switch(config)# show crypto ca<br>certificates admin-ca | Displays the CA certificate information. |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config<br>startup-config | Copies the running configuration to the startup configuration. |

# Deleting RSA Key Pairs from a Cisco NX-OS Device

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

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 2 | **crypto key zeroize rsa** *label*<br><br>**Example:**<br>switch(config)# crypto key zeroize rsa<br>MyKey | Deletes the RSA key pair. |
| Step 3 | **exit**<br><br>**Example:**<br>switch(config)# exit<br>switch# | Exits configuration mode. |
| Step 4 | (Optional) **show crypto key mypubkey rsa**<br><br>**Example:**<br>switch# show crypto key mypubkey rsa | Displays the RSA key pair configuration. |

| | Command or Action | Purpose |
|---|---|---|
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Generating Certificate Requests, on page 183

# Verifying the PKI Configuration

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

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

# Configuration Examples for PKI

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

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

# Configuring Certificates on a Cisco NX-OS Device

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

**Procedure**

**Step 1** Configure the device FQDN.

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

**Step 2**    Configure the DNS domain name for the device.

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

**Step 3**    Create a trust point.

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

**Step 4**    Create an RSA key pair for the device.

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

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

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

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

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

```
Device-1(config)# crypto ca authenticate myCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRRbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
```

```
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
Do you accept this certificate? [yes/no]:y

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

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

```
Device-1(config)# crypto ca enroll myCA
 Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password: nbv123
 The subject name in the certificate will be: Device-1.cisco.com
 Include the switch serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address: 10.10.1.1
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl41KY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

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

**Step 10**   Import the identity certificate.

```
Device-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
```

```
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Device-1(config)# exit
Device-1#
```

**Step 11**     Verify the certificate configuration.

**Step 12**     Save the certificate configuration to the startup configuration.

**Related Topics**

# Downloading a CA Certificate

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

**Procedure**

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

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



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

**Step 4**     In the Certificate dialog box, click **Copy to File** and click **OK**.

**Step 5**     From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next**.

**Step 6**     In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.

**Step 7**     In the Certificate Export Wizard dialog box, click **Finish**.

**Step 8** Enter the Microsoft Windows **type** command to display the CA certificate stored in Base-64 (PEM) format.



## Requesting an Identity Certificate

To request an identify certificate from a Microsoft Certificate server using a PKCS#12 certificate signing request (CRS), follow these steps:

**Procedure**

**Step 1**     From the Microsoft Certificate Services web interface, click **Request a certificate** and click **Next**.



**Step 2**     Click **Advanced request** and click **Next**.

**Step 3** Click **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** and click **Next**.



**Step 4** In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next**. The certificate request is copied from the Cisco NX-OS device console.

**Step 5**      Wait one or two days until the certificate is issued by the CA administrator.

*Microsoft* Certificate Services -- Aparna CA      Home

**Certificate Pending**

Your certificate request has been received. However, you must wait for an administrator to issue the certificate you requested.

Please return to this web site in a day or two to retrieve your certificate.

**Note:** You must return with **this** web browser within 10 days to retrieve your certificate

144769

**Step 6**      Note that the CA administrator approves the certificate request.

| Request ID | Binary Request | Request Disposition Message | Request Submission Date | Requester Name | Reques |
|---|---|---|---|---|---|
| 116 | -----BEGIN NE... | Taken Under Submission | 11/12/2005 8:28 AM | SSE-08\IUSR_SS... | |

Certification Authority (Local)
Aparna CA
Revoked Certificates
Issued Certificates
Pending Requests
Failed Requests

All Tasks → Issue / Deny
Refresh
Help

Contains operations that can be performed on the object.

144770

**Step 7**   From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**.



**Step 8**   Choose the certificate request that you want to check and click **Next**.

**Step 9** Click **Base 64 encoded** and click **Download CA certificate**.



**Step 10** In the File Download dialog box, click **Open**.

**Step 11** In the Certificate box, click **Details** tab and click **Copy to File...**. In the Certificate Export Dialog box, click **Base-64 encoded X.509 (.CER)**, and click **Next**.



**Step 12** In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and



click **Next**.

**Step 13**    Click **Finish**.

**Step 14**    Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.



**Related Topics**

# Revoking a Certificate

To revoke a certificate using the Microsoft CA administrator program, follow these steps:

**Procedure**

**Step 1**    From the Certification Authority tree, click **Issued Certificates** folder. From the list, right-click the certificate that you want to revoke.

**Step 2**    Choose **All Tasks > Revoke Certificate**.



**Step 3**    From the Reason code drop-down list, choose a reason for the revocation and click **Yes**.

**Step 4** Click the **Revoked Certificates** folder to list and verify the certificate revocation.



# Generating and Publishing the CRL

To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

**Procedure**

**Step 1**    From the Certification Authority screen, choose **Action > All Tasks > Publish**.



**Step 2**    In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.

# Downloading the CRL

To download the CRL from the Microsoft CA website, follow these steps:

**Procedure**

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



**Step 2**    Click **Download latest certificate revocation list**.

**Step 3**      In the File Download dialog box, click **Save**.



**Step 4**      In the Save As dialog box, enter the destination file name and click **Save**.

**Step 5**    Enter the Microsoft Windows **type** command to display the CRL.



**Related Topics**

Configuring Certificate Revocation Checking Methods, on page 182

# Importing the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

**Procedure**

**Step 1**    Copy the CRL file to the Cisco NX-OS device bootflash.

```
Device-1# copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

**Step 2**    Configure the CRL.

```
Device-1# configure terminal
Device-1(config)# crypto ca crl request myCA bootflash:aparnaCA.crl
Device-1(config)#
```

**Step 3**    Display the contents of the CRL.

```
Device-1(config)# show crypto ca crl myCA
Trustpoint: myCA
```

```
CRL:
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
        Last Update: Nov 12 04:36:04 2005 GMT
        Next Update: Nov 19 16:56:04 2005 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
            keyid:27:28:F2:46:83:1B:AC:23:4C:45:4D:8E:C9:18:50:1
            1.3.6.1.4.1.311.21.1:
                ...
Revoked Certificates:
    Serial Number: 611B09A1000000000002
        Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
        Revocation Date: Aug 16 21:52:29 2005 GMT
    Serial Number: 4CFC2B42000000000004
        Revocation Date: Aug 16 21:52:41 2005 GMT
    Serial Number: 6C699EC2000000000005
        Revocation Date: Aug 16 21:52:52 2005 GMT
    Serial Number: 6CCF7DDC000000000006
        Revocation Date: Jun  8 00:12:04 2005 GMT
    Serial Number: 70CC4FFF000000000007
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 4D9B1116000000000008
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 52A80230000000000009
        Revocation Date: Jun 27 23:47:06 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 5349AD460000000000000A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C00000000000B
        Revocation Date: Jul  4 18:04:01 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE00000000000C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5D3FD52E00000000000D
        Revocation Date: Jun 29 22:07:25 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Key Compromise
Serial Number: 5DAB771300000000000E
        Revocation Date: Jul 14 00:33:56 2005 GMT
    Serial Number: 5DAE53CD00000000000F
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5DB140D3000000000010
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5E2D7C1B000000000011
        Revocation Date: Jul  6 21:12:10 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Cessation Of Operation
Serial Number: 16DB4F8F000000000012
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 261C3924000000000013
```

```
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 262B5202000000000014
        Revocation Date: Jul 14 00:33:10 2005 GMT
    Serial Number: 2634C7F2000000000015
        Revocation Date: Jul 14 00:32:45 2005 GMT
    Serial Number: 2635B00000000000016
        Revocation Date: Jul 14 00:31:51 2005 GMT
    Serial Number: 264850400000000000017
        Revocation Date: Jul 14 00:32:25 2005 GMT
    Serial Number: 2A2763570000000000018
Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 3F88CBF70000000000019
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 6E4B5F5F0000000000001A
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 725B89D80000000000001B
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 735A887800000000001C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 148511C700000000001D
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14A7170100000000001E
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14FC45B500000000001F
        Revocation Date: Aug 17 18:30:42 2005 GMT
    Serial Number: 486CE80B000000000020
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 4CA4A3AA000000000021
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 1AA55C8E00000000002F
        Revocation Date: Sep  5 17:07:06 2005 GMT
    Serial Number: 3F0845DD00000000003F
        Revocation Date: Sep  8 20:24:32 2005 GMT
    Serial Number: 3F619B7E000000000042
        Revocation Date: Sep  8 21:40:48 2005 GMT
    Serial Number: 6313C463000000000052
        Revocation Date: Sep 19 17:37:18 2005 GMT
    Serial Number: 7C3861E3000000000060
        Revocation Date: Sep 20 17:52:56 2005 GMT
    Serial Number: 7C6EE351000000000061
        Revocation Date: Sep 20 18:52:30 2005 GMT
    Serial Number: 0A338EA1000000000074   <-- Revoked identity certificate
        Revocation Date: Nov 12 04:34:42 2005 GMT
    Signature Algorithm: sha1WithRSAEncryption
        0b:cb:dd:43:0a:b8:62:1e:80:95:06:6f:4d:ab:0c:d8:8e:32:
        44:8e:a7:94:97:af:02:b9:a6:9c:14:fd:eb:90:cf:18:c9:96:
        29:bb:57:37:d9:1f:d5:bd:4e:9a:4b:18:2b:00:2f:d2:6e:c1:
        1a:9f:1a:49:b7:9c:58:24:d7:72
```

**Note** The identity certificate for the device that was revoked (serial number 0A338EA1000000000074) is listed at the end.

# Additional References for PKI

This section includes additional information related to implementing PKI.

# Related Documents for PKI

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS Licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

# Standards for PKI

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

# Resource Public Key Infrastructure (RPKI)

RPKI is a globally distributed database that contains information mapping BGP (internet) prefixes to their authorized origin-AS numbers. To validate the origin-AS of BGP paths, routers running BGP can connect to RPKI caches.

The RPKI-Cache-to-Router connectivity can be many-to-many, one RPKI cache can provide origin-AS validation data to multiple routers and one router can be connected to multiple RPKI caches. A router connects to RPKI caches to download information to build a special RPKI database that can be used by BGP to validate the origin-AS numbers for the internet routing table.

The RPKI database is a set of Route-Origin-Attestation (ROA) objects aggregated from the different RPKI caches to which BGP connects. ROA objects provide a mapping between a BGP prefix-block, and an AS number authorized to originate that block.

# RPKI Configuration

RPKI configuration is categorized as:

- commands for connecting to RPKI Caches.
- commands for marking incoming prefixes with RPKI validation state.
- commands for using RPKI validation state in BGP best-path computation.
- commands for dropping out or manipulating prefixes with specific validation states using route-map.

# Commands for connecting to RPKI caches

RPKI cache configuration is done in a new rpki-cache submode under the router-bgp submode. This is like configuring BGP peers under the default VRF. The submode is entered by using the "rpki cache <IP address>" command. When you enter the submode, various parameters for the RPKI cache can be configured.

```
router bgp 100
 rpki cache 147.28.0.11
    description       A description to identify the cache
    shutdown          Shutdown the cache
    transport tcp port Transport port on which cache is listening
    vrf               Vrf in which RPKI cache is reachable
    refresh-interval  Specify periodic wait time between cache poll attempts
    retry-interval    Specify wait time before retrying failed serial or reset query
   expiry-interval   Specify how long to use current data while unable to perform successful
 query
```

**Note**    Unless transport TCP port is explicitly configured, BGP will connect to RPKI cache on RPKI-RTR port 323.

Unless explicitly configured, all intervals will be determined as suggested by the RPKI Cache in End of Data PDU.

# Commands for marking incoming prefixes with RPKI validation state

There are knobs that control the behavior of RPKI prefix validation processing. These knobs can be configured at the address-family level.

- **origin-as validate** - Configured at the address-family level enables eBGP path validation against ROA database. By default, this is disabled.

**Note**    This command has no bearing on iBGP paths. The iBGP paths are not validated against ROA database. The only way to mark path validation state on iBGP paths is receiving the BGP Prefix Origin Validation State Extended Community, and is done by default without configuring any command.

- **origin-as validate signal ibgp** - Configured at the address-family level enables the iBGP signalling of validity state through BGP Prefix Origin Validation State Extended Community.

# Commands for using RPKI validation state in BGP best-path-computation

There are commands to control the behavior of RPKI prefix validation processing. These commands can be configured at the address-family level.

- **bestpath origin-as use-validity** - Configured at the address-family level enables the validity states of BGP paths to affect the path's preference in the BGP bestpath process. By default, this is disabled.

- **bestpath origin-as allow invalid** - Configured at the address-family level allows all "invalid" paths to be considered for BGP bestpath computation (all such paths are not bestpath candidates if best-path origin-as validate is configured). By default, this is disabled.

# Commands for dropping out or manipulating prefixes with specific validation states using route-map

The following is the command for dropping out or manipulating prefixes with specific validation states using route-map:

```
route-map sample1 permit 10
  match rpki {not-found | invalid | valid}
```

The parameters of the match rpki command are described as follows:

- `not-found` - This origin-AS is unknown in the RPKI database.

- `invalid` - This is an invalid origin-AS in the RPKI database.

  `valid` - This is a valid origin-AS in the RPKI database.

This match clause is relevant for inbound route-maps only.

For iBGP learnt paths, the incoming BGP Prefix Origin Validation State Extended Community in the update will be compared against this route-map clause.

For eBGP learnt paths, the validation state obtained by ROA database lookup will be compared against this route-map clause.

While prefixes marked as validation-state invalid are rendered ineffective by not being considered for best-path computation in BGP, an administrator may decide to drop such prefixes altogether to save system memory. The following inbound route-map is recommended for this purpose:

```
route-map sample deny 10
match rpki invalid
route-map sample permit 20
```

# RPKI Show Commands

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

| Command | Purpose |
|---|---|
| **show bgp rpki summary** | Displays an overview of RPKI statistics including the number of RPKI caches. |

| Command | Purpose |
|---|---|
| **show bgp rpki table** {**ipv4** \| **ipv6**} {**IP address/masklength**} | Displays information about the current RPKI ROA database. With no options specified, the command shows the IPv4 ROA database. With the IPv6 option (show bgp rpki table ipv6), the command shows the IPv6 ROA database. ROAs that are received from a cache that is temporarily down (due to connectivity issues, for example) are displayed with (*). These ROAs will be removed from the RPKI database if the cache session does not establish within the purge-time for that cache. |
| | If an ROA prefix-block is specified after the table show command (for example, show bgp rpki table 67.21.36.0/24 max 24), then that specific ROA entry is displayed in detail, if the ROA exists. |
| | **Note**  One ROA (IP address/min-max) can have multiple origin ASs and can be sourced from multiple caches. |
| **show bgp rpki cache** {**IP address**} | Displays a summary listing of all the caches that are configured and their parameters, such as **show bgp summary**. |
| | If a cache IP address is specified with the previous command, then detailed information is shown for that cache. |
| **show bgp** {**ipv4 unicast** \| **ipv6 unicast**} **origin-as validity-state** {**valid** \| **invalid** \| **unknown**} | Displays information about BGP. This command has new options to filter the BGP table output based on path (validation_state). Specify a validity state (valid, invalid, or unknown) with this command to filter the relevant information from the BGP table, and only the BGP paths matching that validity-state are displayed. |

# RPKI Clear Commands

The following is the RPKI Clear command:

- **clear bgp rpki cache \*** - This command resets the transport sessions of all configured RPKI caches and immediately purges the RPKI database of all IPv4 and IPv6 ROAs received from all caches.

# RPKI Debug and Event History Commands

The following are the RPKI Debug and Event History commands:

- **debug bgp rpki** - This command turns on debugging for all RPKI related operations excluding prefix-validation. This includes debugging events such as RPKI cache connectivity, protocol state-machine for the RPKI caches, and RPKI database events such as ROA insertion or deletion.

- **sh bgp event-history rpki** - This command dumps high level information about RPKI.

# Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on Cisco NX-OS devices.

This chapter includes the following sections:

## About User Accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the Cisco NX-OS device. RBAC allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

## User Accounts

You can configure up to a maximum of 256 user accounts. By default, the user account does not expire unless you explicitly configure it to expire. The expire option determines the date when the user account is disabled.

The following words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, root, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.

**Note** User passwords are not displayed in the configuration files.

⚠️

**Caution**  Usernames must begin with an alphanumeric character and can contain only these special characters: ( + = . _ \ -). The # and ! symbols are not supported. If the username contains characters that are not allowed, the specified user is unable to log in.

# Characteristics of Strong Passwords

A strong password has the following characteristics:

✎

**Note**  Special characters, such as the dollar sign ($) or the percent sign (%), can be used in Cisco Nexus device passwords.

- Is at least eight characters long
- Does not contain many consecutive characters (such as abcd)
- Does not contain many repeating characters (such as aaabbb)
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

✎

**Note**  Clear text passwords cannot include these special characters at the beginning of the password: quotation marks (" or '), vertical bars (|), or right angle brackets (>). If a password is trivial (such as a short, easy-to-decipher password), the Cisco NX-OS software will reject your password configuration if password-strength checking is enabled. Be sure to configure a strong password as shown in the sample configuration. Passwords are case-sensitive.

✎

**Note**  All printable ASCII characters are supported in the password string if they are enclosed in quotation marks.

**Related Topics**

# User Roles

User roles contain rules that define the operations allowed for the user who is assigned the role. Each user role can contain multiple rules, and each user can have multiple roles. For example, if role1 allows access only to configuration operations, and role2 allows access only to debug operations, then users who belong to both role1 and role2 can access configuration and debug operations. You can also limit access to specific virtual routing and forwarding instances (VRFs), VLANs, and interfaces.

The Cisco NX-OS software provides the following user roles:

- network-admin—Complete read-and-write access to the entire Cisco NX-OS device

- network-operator or vdc-operator—Complete read access to the entire Cisco NX-OS device

**Note**
- The Cisco Nexus 9000 Series switches do not support multiple VDCs; however, the vdc-operator role is available and has the same privileges and limitations as the network-operator role.

- The Cisco Nexus 9000 Series switches support a single VDC due to which the vdc-admin has the same privileges and limitations as the network-admin.

**Note** You cannot change the user roles.

**Note** Some **show** commands may be hidden from network-operator users. In addition, some non-**show** commands (such as **telnet**) may be available for this user role.

By default, the user accounts without an administrator role can access only the **show**, **exit**, **end**, and **configure terminal** commands. You can add rules to allow users to configure features.

**Note** If you belong to multiple roles, you can execute a combination of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose a user has RoleA, which denied access to the configuration commands. However, the user also has RoleB, which has access to the configuration commands. In this case, the user has access to the configuration commands.

# User Role Rules

The rule is the basic element of a role. A rule defines what operations the role allows the user to perform. You can apply rules for the following parameters:

**Command**
A command or group of commands defined in a regular expression.
**Feature**
A command or group of commands defined in a regular expression.

**Feature group**
> Default or user-defined group of features.

**OID**
> An SNMP object identifier (OID).

The command, feature, and feature group parameters create a hierarchical relationship. The most basic control parameter is the command. The next control parameter is the feature, which represents all commands associated with the feature. The last control parameter is the feature group. The feature group combines related features and allows you to easily manage the rules. The Cisco NX-OS software also supports the predefined feature group L3 that you can use.

SNMP OID is supported for RBAC. You can configure a read-only or read-and-write rule for an SNMP OID.

You can configure up to 256 rules for each role. The user-specified rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

# Guidelines and Limitations for User Accounts and RBAC

User accounts and RBAC have the following configuration guidelines and limitations:

- You can add up to 256 rules to a user role.

- You can add up to 64 user-defined feature groups in addition to the default feature group, L3.

- You can configure up to 256 users.

- You can assign a maximum of 64 user roles to a user account.

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

- You cannot delete the default admin and SNMP user accounts.

- You cannot remove the default user roles from the default admin user accounts.

- The network-operator role cannot run the **show running-config** and **show startup-config** commands.

- The Cisco Nexus 9000 Series switches support a single VDC due to which the vdc-admin has the same privileges and limitations as the network-admin.

- As per the AAA policy, if a role is associated as a last role with an user, then that role cannot be deleted until it is disassociated from that user.

> ✎
>
> **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.

- Beginning with Cisco NX-OS Release 10.2(2)F, a new desynchronization CLI is introduced to provide you an option to disable the user synchronization between the SNMP and the security components. For more information, refer to the *Configuring SNMP* chapter in the *System Management Configuration Guide*.

For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- When the desynchronization CLI is enabled, remote users will not be synced to SNMP database.

- The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a) will not have a corresponding SNMPv3 profile when the desynchronization CLI is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization CLI along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

- Beginning with Cisco NX-OS Release 10.3(1)F, the type 8 and type 9 password hash is supported on Cisco Nexus 9000 Series switches.

**Note**     Type 8 and type 9 cannot be downgraded though type 5 supports downward compatibility.

- Beginning with Cisco NX-OS Release 10.3(1)F, the consecutive characters check in passwords is supported on Cisco Nexus 9000 Series switches.

# Default Settings for User Accounts and RBAC

This table lists the default settings for user accounts and RBAC parameters.

*Table 13: Default User Accounts and RBAC Parameters*

| Parameters | Default |
|---|---|
| User account password | Undefined |
| User account expiry date | None |
| User account role | Network-operator if the creating user has the network-admin role |
| Default user role | Network-operator |
| Interface policy | All interfaces are accessible |
| VLAN policy | All VLANs are accessible |
| VRF policy | All VRFs are accessible |
| Feature group | L3 |

# Enabling Password-Strength Checking

You can enable password-strength checking which prevents you from creating weak passwords for user accounts.

✎

**Note**    When you enable password-strength checking, the Cisco NX-OS software does not check the strength of existing passwords.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal`<br>`switch(config)#``` | Enters global configuration mode. |
| **Step 2** | **password strength-check**<br><br>**Example:**<br><br>`switch(config)# password strength-check` | Enables password-strength checking. The default is enabled.<br><br>You can disable password-strength checking by using the **no** form of this command. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>```switch(config)# exit`<br>`switch#``` | Exits global configuration mode. |
| **Step 4** | (Optional) **show password strength-check**<br><br>**Example:**<br><br>`switch# show password strength-check` | Displays the password-strength check configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```switch# copy running-config`<br>`startup-config``` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Enabling Consecutive Characters Check in Passwords

The password sequence keyboard length and alphabet length are imposed restrictions as they are vulnerable to attacks.

Following length limit of password string sequences are imposed on the password:

- Number of repeated characters based on configurable value (aaaa, bbbb, etc)

- Number of consecutive alphabetical/numeric sequence characters (abcd, 1234,..)

- Number of consecutive keyboard sequence characters (qwer, asdf..)

This procedure describes how to configure the limits for password sequences.

**Procedure**

|        | Command or Action | Purpose |
|--------|-------------------|---------|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enter configuration mode. |
| **Step 2** | **[no] userpassphrase sequence alphabet length** *Value*<br><br>**Example:**<br><br>`switch(config)#userpassphrase sequence`<br>`alphabet length 4` | Configures the limit of sequence alphabet length. The range for sequence alphabet length is 2-10.<br><br>Example: **userpassphrase sequence alphabet length** *4*<br><br>**username user password** *AbcDe19jd*<br><br>Password characters are sequential, hence cannot be accepted.<br><br>The **no** option disables the alphabet sequence check. |
| **Step 3** | **[no] userpassphrase sequence keyboard length** *Value*<br><br>**Example:**<br><br>`switch(config)# userpassphrase sequence`<br>` keyboard length 4` | Configures the limit of sequence keyboard length. The range for sequence alphabet length is 2-10.<br><br>Example: **userpassphrase sequence keyboard length** *4*<br><br>**username user password** *CvBnmwu204*<br><br>Password characters are sequential, hence cannot be accepted.<br><br>The **no** option disables the keyboard sequence check. |

# Configuring User Accounts

You can create a maximum of 256 user accounts on a Cisco NX-OS device. User accounts have the following attributes:

- Username

- Password

- Expiry date

- User roles

You can enter the password in clear text format or encrypted format. The Cisco NX-OS password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption.

SHA256 is the hashing algorithm used for password encryption. As a part of the encryption, a 5000 iteration of 64-bit SALT is added to the password.

SHA256 is the default hashing algorithm used for password encryption. To generate a hash for type 8 and type 9 password, you must provide PBKDF2/SCRYPT option along with clear text password.

User accounts can have a maximum of 64 user roles. The user can determine what commands are available by using the command-line interface (CLI) context sensitive help utility.

---

**Note**   Changes to user account attributes do not take effect until the user logs in and creates a new session.

---

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | (Optional) **show role**<br><br>**Example:**<br><br>`switch(config)# show role` | Displays the user roles available. You can configure other user roles, if necessary. |
| **Step 3** | **username** *user-id* [**password** [**0** \| **5** \| **8** \| **9**] *password* [**pbkdf2** \| **scrypt**]] [**expire** *date*] [**role** *role-name*]<br><br>**Example:**<br><br>`switch(config)# username NewUser password` `4Ty18Rnt` | Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters for both local and remote users. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.<br><br>Usernames must begin with an alphanumeric character.<br><br>The default password is undefined.<br><br>- The **0** option indicates that the password is clear text |

| | Command or Action | Purpose |
|---|---|---|
| | | • The **5** option indicates that the password is SHA-256 hashed. |
| | | • The **8** option indicates that the password is PBKDF2 hashed. |
| | | • The **9** option indicates that the password is Scrypt hashed. |
| | | The default option is **0** (clear text). |
| | | **Note**    The pbkdf2/scrypt keywords are optional and are note stored in running configurations. |
| | | **Note**    If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device. |
| | | **Note**    If you create a user account with the encrypted password option, the corresponding SNMP user will not be created. |
| | | **Note**    When the desynchronization CLI is enabled, if you create a user account, the corresponding SNMP user will not be created. |
| | | The **expire** *date* option format is YYYY-MM-DD. The default is no expiry date. |
| | | User accounts can have a maximum of 64 user roles. |
| **Step 4** | **username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** \| **rsa**}<br><br>**Example:**<br>`switch(config)# username NewUser ssh-cert-dn "/CN = NewUser, OU = Cisco Demo, O = Cisco, C = US" rsa`<br><br>**Example:**<br>`switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa` | Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively. |
| **Step 5** | **exit**<br><br>**Example:** | Exits global configuration mode. |

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

**Related Topics**

# Configuring Roles

This section describes how to configure user roles.

# Creating User Roles and Rules

You can configure up to 64 user roles. Each user role can have up to 256 rules. You can assign a user role to more than one user account.

The rule number that you specify determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

When processing an RBACL for a match, a partial match does not stop the evaluation process. Evaluation continues through each rule until an exact match is found. If no exact match is found, the most precise rule in the list will be chosen for the result. Also, if a permit and deny rule exists for the same match logic, the higher numbered rule (evaluated first) will be chosen for the result.

**Note**     Regardless of the read-write rule configured for a user role, some commands can be executed only through the predefined network-admin role.

**Before you begin**

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **role name** *role-name*<br><br>**Example:**<br><br>`switch(config)# role name UserA`<br>`switch(config-role)#` | Specifies a user role and enters role configuration mode. The *role-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 16 characters. |
| **Step 3** | **rule** *number* {**deny** \| **permit**} **command** *command-string*<br><br>**Example:**<br><br>`switch(config-role)# rule 1 deny command`<br>` clear users` | Configures a command rule.<br><br>The *command-string* argument can contain spaces and regular expressions. For example, interface ethernet includes all Ethernet interfaces.<br><br>Repeat this command for as many rules as needed. |
| **Step 4** | **rule** *number* {**deny** \| **permit**} {**read** \| **read-write**}<br><br>**Example:**<br><br>`switch(config-role)# rule 2 deny`<br>`read-write` | Configures a read-only or read-and-write rule for all operations. |
| **Step 5** | **rule** *number* {**deny** \| **permit**} {**read** \| **read-write**} **feature** *feature-name*<br><br>**Example:**<br><br>`switch(config-role)# rule 3 permit read`<br>` feature router-bgp` | Configures a read-only or read-and-write rule for a feature.<br><br>Use the **show role feature** command to display a list of features.<br><br>Repeat this command for as many rules as needed. |
| **Step 6** | **rule** *number* {**deny** \| **permit**} {**read** \| **read-write**} **feature-group** *group-name*<br><br>**Example:**<br><br>`switch(config-role)# rule 4 deny`<br>`read-write feature-group L3` | Configures a read-only or read-and-write rule for a feature group.<br><br>Use the **show role feature-group** command to display a list of feature groups.<br><br>Repeat this command for as many rules as needed. |
| **Step 7** | **rule** *number* {**deny** \| **permit**} {**read** \| **read-write**} **oid** *snmp_oid_name*<br><br>**Example:**<br><br>`switch(config-role)# rule 5 deny`<br>`read-write oid 1.3.6.1.2.1.1.9` | Configures a read-only or read-and-write rule for an SNMP object identifier (OID). You can enter up to 32 elements for the OID. This command can be used to allow SNMP-based performance monitoring tools to poll devices but restrict their access to system-intensive |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | branches such as the IP routing table, MAC address tables, specific MIBs, and so on. |
| | | **Note** The deepest OID can be at the scalar level or at the table root level. |
| | | Repeat this command for as many rules as needed. |
| **Step 8** | (Optional) **description** *text*<br><br>**Example:**<br>`switch(config-role)# description This role does not allow users to use clear commands` | Configures the role description. You can include spaces in the description. |
| **Step 9** | **exit**<br><br>**Example:**<br>`switch(config-role)# exit`<br>`switch(config)#` | Exits role configuration mode. |
| **Step 10** | (Optional) **show role**<br><br>**Example:**<br>`switch(config)# show role` | Displays the user role configuration. |
| **Step 11** | (Optional) **show role** {**pending** \| **pending-diff**}<br><br>**Example:**<br>`switch(config)# show role pending` | Displays the user role configuration pending for distribution. |
| **Step 12** | (Optional) **role commit**<br><br>**Example:**<br>`switch(config)# role commit` | Applies the user role configuration changes in the temporary database to the running configuration. |
| **Step 13** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Creating Feature Groups

You can create custom feature groups to add to the default list of features provided by the Cisco NX-OS software. These groups contain one or more of the features. You can create up to 64 feature groups.

**Note** You cannot change the default feature group L3.

**Before you begin**

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **role feature-group name** *group-name*<br><br>**Example:**<br>`switch(config)# role feature-group name`<br>`GroupA`<br>`switch(config-role-featuregrp)#` | Specifies a user role feature group and enters role feature group configuration mode.<br><br>The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters. |
| **Step 3** | **feature** *feature-name*<br><br>**Example:**<br>`switch(config-role-featuregrp)# feature`<br>`radius` | Specifies a feature for the feature group.<br><br>Repeat this command for as many features as needed.<br><br>**Note**   Use the **show role component** command to display a list of features. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config-role-featuregrp)# exit`<br>`switch(config)#` | Exits role feature group configuration mode. |
| **Step 5** | (Optional) **show role feature-group**<br><br>**Example:**<br>`switch(config)# show role feature-group` | Displays the role feature group configuration. |
| **Step 6** | (Optional) **show role** {**pending** | **pending-diff**}<br><br>**Example:**<br>`switch(config)# show role pending` | Displays the user role configuration pending for distribution. |
| **Step 7** | (Optional) **role commit**<br><br>**Example:**<br>`switch(config)# role commit` | Applies the user role configuration changes in the temporary database to the running configuration. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Changing User Role Interface Policies

You can change a user role interface policy to limit the interfaces that the user can access. By default, a user role allows access to all interfaces.

### Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **role name** *role-name*<br><br>**Example:**<br>`switch(config)# role name UserA`<br>`switch(config-role)#` | Specifies a user role and enters role configuration mode. |
| **Step 3** | **interface policy deny**<br><br>**Example:**<br>`switch(config-role)# interface policy`<br>`deny`<br>`switch(config-role-interface)#` | Enters role interface policy configuration mode. |
| **Step 4** | **permit interface** *interface-list*<br><br>**Example:**<br>`switch(config-role-interface)# permit`<br>`interface ethernet 2/1-4` | Specifies a list of interfaces that the role can access.<br><br>Repeat this command for as many interfaces as needed. |
| **Step 5** | **exit**<br><br>**Example:**<br>`switch(config-role-interface)# exit`<br>`switch(config-role)#` | Exits role interface policy configuration mode. |
| **Step 6** | (Optional) **show role**<br><br>**Example:**<br>`switch(config-role)# show role` | Displays the role configuration. |
| **Step 7** | (Optional) **show role** {**pending** \| **pending-diff**}<br><br>**Example:**<br>`switch(config-role)# show role pending` | Displays the user role configuration pending for distribution. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | (Optional) **role commit**<br><br>**Example:**<br>`switch(config-role)# role commit` | Applies the user role configuration changes in the temporary database to the running configuration. |
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-role)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Changing User Role VLAN Policies

You can change a user role VLAN policy to limit the VLANs that the user can access. By default, a user role allows access to all VLANs.

**Before you begin**

Create one or more user roles.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **role name** *role-name*<br><br>**Example:**<br>`switch(config)# role name UserA`<br>`switch(config-role)#` | Specifies a user role and enters role configuration mode. |
| **Step 3** | **vlan policy deny**<br><br>**Example:**<br>`switch(config-role)# vlan policy deny`<br>`switch(config-role-vlan)#` | Enters role VLAN policy configuration mode. |
| **Step 4** | **permit vlan** *vlan-list*<br><br>**Example:**<br>`switch(config-role-vlan)# permit vlan`<br>`1-4` | Specifies a range of VLANs that the role can access.<br><br>Repeat this command for as many VLANs as needed. |
| **Step 5** | **exit**<br><br>**Example:** | Exits role VLAN policy configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-role-vlan)# exit`<br>`switch(config-role)#` | |
| Step 6 | (Optional) **show role**<br><br>**Example:**<br>`switch(config)# show role` | Displays the role configuration. |
| Step 7 | (Optional) **show role** {**pending** \| **pending-diff**}<br><br>**Example:**<br>`switch(config-role)# show role pending` | Displays the user role configuration pending for distribution. |
| Step 8 | (Optional) **role commit**<br><br>**Example:**<br>`switch(config-role)# role commit` | Applies the user role configuration changes in the temporary database to the running configuration. |
| Step 9 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-role)# copy running-config`<br>` startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Creating User Roles and Rules, on page 228

# Changing User Role VRF Policies

You can change a user role VRF policy to limit the VRFs that the user can access. By default, a user role allows access to all VRFs.

**Before you begin**

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **role name** *role-name*<br><br>**Example:**<br>`switch(config)# role name UserA`<br>`switch(config-role)#` | Specifies a user role and enters role configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **vrf policy deny**<br><br>**Example:**<br>`switch(config-role)# vrf policy deny`<br>`switch(config-role-vrf)#` | Enters role VRF policy configuration mode. |
| **Step 4** | **permit vrf** *vrf-name*<br><br>**Example:**<br>`switch(config-role-vrf)# permit vrf vrf1` | Specifies the VRF that the role can access.<br><br>Repeat this command for as many VRFs as needed. |
| **Step 5** | **exit**<br><br>**Example:**<br>`switch(config-role-vrf)# exit`<br>`switch(config-role)#` | Exits role VRF policy configuration mode. |
| **Step 6** | (Optional) **show role**<br><br>**Example:**<br>`switch(config-role)# show role` | Displays the role configuration. |
| **Step 7** | (Optional) **show role** {**pending** \| **pending-diff**}<br><br>**Example:**<br>`switch(config-role)# show role pending` | Displays the user role configuration pending for distribution. |
| **Step 8** | (Optional) **role commit**<br><br>**Example:**<br>`switch(config-role)# role commit` | Applies the user role configuration changes in the temporary database to the running configuration. |
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-role)# copy running-config`<br>` startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Creating User Roles and Rules, on page 228

# About No Service Password-Recovery

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

# Enabling No Service Password-Recovery

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

### Before you begin

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

### Procedure

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| **Step 2** | **no service password-recovery**<br><br>**Example:**<br><br>```<br>switch(config)# no service<br>password-recovery<br>WARNING: Executing this command will<br>disable the password recovery mechanism.<br> Do not execute this command without<br>another plan for password recovery. Are<br> you sure you want to continue? (y/n) :<br> [y] y<br>switch(config)# copy run start<br>[#######################################]<br> 100%<br>Copy complete, now saving to disk (please<br> wait)...<br>Copy complete.<br>``` | Disables the password recovery mechanism. |
| **Step 3** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```<br>switch# copy running-config<br>startup-config<br>``` | Copies the running configuration to the startup configuration. |
| **Step 4** | **Reload**<br><br>**Example:**<br><br>```<br>switch(config)# Reload<br>This command will reboot the system.<br>(y/n)?  [n] y<br>2018 Jun 26 16:23:19 BAR %$ VDC-1 %$<br>%PLATFORM-2-PFM_SYSTEM_RESET: Manual<br>system restart from Command Line<br>Interface<br><br>CISCO SWITCH Ver 8.34<br>``` |  |

| | Command or Action | Purpose |
|---|---|---|
| | ```
CISCO SWITCH Ver 8.34
Manual system restart from Command Line
 Interface
writing reset reason 9,
..
..

switch(boot)# config t
Enter configuration commands, one per
line.  End with CNTL/Z.
switch(boot)(config)# admin-password
Abcd!123$
ERROR: service password-recovery
disabled. Cannot change password!
switch(boot)(config)#
``` | |
| **Step 5** | **exit**<br><br>**Example:**<br><br>```
switch(config)# exit
switch#
``` | Exits global configuration mode. |
| **Step 6** | (Optional) **show user-account**<br><br>**Example:**<br><br>```
switch# show user-account
``` | Displays the role configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```
switch# copy running-config
startup-config
``` | Copies the running configuration to the startup configuration. |

# Verifying User Accounts and RBAC Configuration

To display user account and RBAC configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show cli syntax roles network-admin** | Displays the syntax of the commands that the network-admin role can use. |
| **show cli syntax roles network-operator** | Displays the syntax of the commands that the network-operator role can use. |
| **show role** | Displays the user role configuration. |
| **show role feature** | Displays the feature list. |

| Command | Purpose |
| --- | --- |
| **show role feature-group** | Displays the feature group configuration. |
| **show startup-config security** | Displays the user account configuration in the startup configuration. |
| **show running-config security** [**all**] | Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts. |
| **show user-account** | Displays user account information. |

# Configuration Examples for User Accounts and RBAC

The following example shows how to configure a user role:

```
role name User-role-A
  rule 2 permit read-write feature bgp
  rule 1 deny command clear *
```

The following example shows how to create a user role that can configure an interface to enable and show BGP and show EIGRP:

```
role name iftest
   rule 1 permit command config t; interface *; bgp *
   rule 2 permit read-write feature bgp
   rule 3 permit read feature eigrp
```

In the above example, rule 1 allows you to configure BGP on an interface, rule 2 allows you to configure the **config bgp** command and enable the exec-level **show** and **debug** commands for BGP, and rule 3 allows you to enable the exec-level **show** and **debug eigrp** commands.

The following example shows how to configure a user role that can configure only a specific interface:

```
role name Int_Eth2-3_only
  rule 1 permit command configure terminal; interface *
  interface policy deny
    permit interface Ethernet2/3
```

The following example shows how to configure a user role feature group:

```
role feature-group name Security-features
  feature radius
  feature tacacs
  feature aaa
  feature acl
  feature access-list
```

The following example shows how to configure a user account:

```
username user1 password A1s2D4f5 role User-role-A
```

The following example shows how to add an OID rule to restrict access to part of the OID subtree:

```
role name User1
  rule 1 permit read feature snmp
  rule 2 deny read oid 1.3.6.1.2.1.1.9
show role name User1

Role: User1
  Description: new role
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  ----------------------------------------------------------------
  Rule     Perm     Type        Scope            Entity
  ----------------------------------------------------------------
  2        deny     read        oid              1.3.6.1.2.1.1.9
  1        permit   read        feature          snmp
```

The following example shows how to give write permission to a specified OID subtree:

```
role name User1
rule 3 permit read-write oid 1.3.6.1.2.1.1.5
show role name User1

Role: User1
  Description: new role
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  ----------------------------------------------------------------
  Rule     Perm     Type        Scope            Entity
  ----------------------------------------------------------------
  3        permit   read-write  oid              1.3.6.1.2.1.1.5
  2        deny     read        oid              1.3.6.1.2.1.1.9
  1        permit   read        feature          snmp
```

# Additional References for User Accounts and RBAC

This section includes additional information related to implementing user accounts and RBAC.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS Licensing | *Cisco NX-OS Licensing Guide* |
| VRF configuration | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

**Standards**

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

**MIBs**

| MIBs | MIBs Link |
|---|---|
| MIBs related to user accounts and RBAC | To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html |

# Configuring 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on Cisco NX-OS devices.

This chapter includes the following sections:

## About 802.1X

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

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

## Device Roles

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

**Figure 5: 802.1X Device Roles**



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.

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

If the Cisco NX-OS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the Cisco NX-OS device does not unauthorize the client connected to the interface. When reauthentication occurs, the Cisco NX-OS device uses 802.1X authentication as the preferred reauthentication process.

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

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

MAC authentication bypass interacts with the following features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.

- Port security—You cannot configure 802.1X authentication and port security on the same Layer 2 ports.

- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

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

The Cisco Nexus 9000 Series switches supports dynamic VLAN assignment. After the 802.1x authentication or MAB is completed; before bringing up the port, you may want to (as part of authorization) allow the peer/host to be placed into a particular VLAN based as a result of the authentication. The RADIUS server

typically indicates the desired VLAN by including tunnel attributes within the Access-Accept message. This procedure of getting the VLAN an binding it to the port constitutes to Dynamic VLAN assignment.

# VLAN Assignment from RADIUS

After authentication is completed either through 802.1X 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 Topology

The 802.1X port-based authentication supports point-to-point topology.

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

## About Per-User DACLs

From Cisco NX-OS Release 10.2(1)F, you can download per-user dynamic access control lists (DACLs) from the Cisco ISE Server as policy enforcement after authentication using IEEE 802.1X.

Per-user DACLs can be configured to provide different levels of network access and service to an 802.1X-authenticated user. When the RADIUS server authenticates a user that is connected to an 802.1X port, it retrieves the ACL attributes based on the user identity and sends them to the switch. The switch applies the attributes to the 802.1X port for the duration of the user session. The switch removes the per-user DACL configuration whenever the session is terminated or if the authentication failed.

RADIUS supports per-user attributes, including vendor-specific attributes. These vendor-specific attributes (VSAs) are in the octet-string format and are passed to the switch during the authentication process. The VSAs used for per-user DACLs are `inacl#<n>` for the ingress direction, where the value of `n` is from 1 to 32. The syntax is as follows:

`ip:inacl#<n>=`**`permit | deny`** `[`*`protocol`*`] [`*`source_subnet`*`] [`*`dest_subnet`*`] [`*`operator`*`][`*`port`*`]`

Example 1: `ip:inacl#1=permit udp any any eq 5555`

Example 2: `ip:inacl#2=deny udp any any eq 6666`

The switch supports VSAs only in the ingress direction.

## Critical Authentication

From Cisco NX-OS Release 10.1(1), the 802.1X critical authentication on a port, accommodates 802.1X users that failed authentication when RADIUS servers in their ISP domain weren't reachable. The critical authentication feature is supported when 802.1X authentication is performed only through RADIUS or ISE servers. If an 802.1X user fails RADIUS authentication, it's still allowed to access the network. You can achieve this by using the **dot1x authentication event server dead action authorize** command. Use the **no** command to disable this feature.

Beginning with Cisco NX-OS Release 10.5(2)F, 802.1x feature and critical authentication feature are supported on multi-domain ports.

# About 802.1X for Voice VLAN

## Overview of 802.1X for Voice VLAN

The IEEE 802.1X Voice VLAN feature enables multi-domain 802.1X authentication on a single port, providing authentication support for both VoIP phones and data client connected to them. This feature allows you to configure a special access port associated with two VLAN identifiers: one dedicated to voice traffic and the other to data traffic. This setup requires support for multi-domain host mode, accommodating one voice client and one data client on the same port.

Users would utilize this feature to ensure secure and efficient network traffic management for both voice and data devices connected through a single access port. By segregating voice and data traffic into separate VLANs, network administrators can ensure high-quality VoIP communication while maintaining robust security and authentication for both types of devices.

# Functionalities of 802.1X for Voice VLAN

• Enables multi-domain 802.1X authentication on a single port.

• A new host mode (multi-domain) supports one voice device (typically an IP phone) and one data device (typically a PC) on a single port.

• Runs a single authenticator on the port capable of distinguishing users based on their identity and placing them in their appropriate domains (different VLANs for data and voice).

• Maintains separation between data traffic and voice traffic on the same port.



# Message Exchange of 802.1X for Voice VLAN

• **VoIP Phone Connection**: A VoIP phone is connected to a switch where 802.1X is configured.

• **VoIP Phone Authentication**: The VoIP phone can authenticate using either EAP or MAB.

• **Voice Device Recognition**: The VoIP phone is recognized as a voice device by the RADIUS server, and 802.1X secures the phone in the voice VLAN.

• **Data Device Connection**: A data device (e.g., a laptop) is connected to the switch using the VoIP phone.

• **Data Device Authentication**: The data device triggers 802.1X authentication and gets authorized by the RADIUS server in the data VLAN.

## About DACL

Dynamic ACL (DACL) is a single ACL that contains permissions of what users and groups can access. It restricts access to the dot1x MAB client. The DACL policy is pushed from the Cisco ISE server to blacklist a MAC address. It applies ACLs on the blacklisted MAC, enabling limited access to the MAB. A single DACL supports all blacklisted MAB clients.

In Cisco NX-OS Release 9.3(5), the DACL is preconfigured on the Cisco Nexus switches.

## Prerequisites for 802.1X

• Cisco Nexus Release 7.0(3)I7(1) software.

The following prerequisites are required for 802.1X Port-based Authentication with EAP-TLS profile:

• PKI Infra is responsible for providing the certificate management for EAP-TLS. This includes

  • Generating an RSA key-pair

  • Creation of certificate trustpoint

  • Authenticating the CA

• 802.1X needs a remote EAP server such as ISE on the device to provide the EAP-TLS. Local authentication server is not supported.

- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.

- AAA server reachability: For the switches to mutually authenticate each other.

- As the switches do mutual authentication, both of them must have proper AAA configurations and AAA connectivity.

# 802.1X Guidelines and Limitations

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

- When you upgrade the Cisco Nexus Series switch to Cisco NX-OS Release 9.2(1) using the (dispruptive/non-disruptive) In-Service Software Upgrades (ISSU), you must first disable 802.1x using the **no feature dot1x** command and then enable it using the **feature dot1x** command for multi-authentication to work.

- Beginning with Cisco NX-OS Release 9.2(1), multi-authentication mode is enabled on an 802.1X port. Dynamic VLAN assignment occurs successfully for the first authenticated host. Subsequent authorized (based on user credentials) data hosts are considered successfully authenticated, provided either they have no VLAN assignment or have a VLAN assignment matching the first successfully authenticated host on the port. This ensures that all successfully authenticated hosts on a port are members of the same VLAN. Flexibility of Dynamic VLAN assignment is only provided to the first authenticated host.

- Beginning with Cisco NX-OS Release 9.2(3), 802.1X port-based authentication is supported on FEX-ST and host interface (HIF) ports. IEEE 802.1X port-based authentication support applies to both straight-through and dual-homed FEX.

- Cisco Nexus 9000 Series switches do not support 802.1X on the following:

    - Transit topology set ups

    - vPC ports

    - PVLAN ports

    - L3 (routed) ports

    - Port security

    - Ports that are enabled with CTS and MACsec PSK.

    - 802.1X with LACP port-channels.

**Note**    802.1X supports static port-channels.

**Note**    Disable 802.1X on vPC ports and all unsupported features.

- The Cisco NX-OS software supports 802.1X authentication only on physical ports.

- Dynamic VLAN assignment is supported only on Cisco Nexus 9300-FX/EX/FX2 Platform switches.

- The Cisco NX-OS software does not work with the CTS or the MACsec PSK features. Global "mac-learn disable" and 802.1X feature are mutually exclusive and cannot be configured together.

- 802.1X is mutually exclusive with the IP Source Guard and uRPF features and cannot be configured together. When you upgrade the Cisco Nexus Series switch to Cisco NX-OS Release 9.2(3), you must disable one of these features.

- During a switch reload, 802.1X does not generate RADIUS accounting stops.

- 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

    - Guest VLANs

- In order to prevent reauthentication of inactive sessions, use the authentication timer inactivity command to set the inactivity timer to an interval shorter than the reauthentication interval set with the authentication timer reauthenticate command.

- A security violation occurs when the same MAC is learned on a different VLAN with 802.1X enabled on the interface.

- Configuring mac learn disable with 802.1X enabled on a DME enabled platform does not display the error messages.

- In Cisco Nexus Release 9.2(1), tagged EAPOL frames are processed although the VLAN is not configured on the interface and the authentication is successful on the interface for the client.

- Secure mac learned on the orphan port is not synced on the vPC peer.

- Beginning with Cisco NX-OS Release 9.2(1), the MAC authentication bypass is supported on Cisco Nexus 9300-EX/FX/FX2 TOR switches.

- Beginning with Cisco NX-OS Release 9.3(5), 802.1X is supported on Cisco Nexus 9300-FX3 platform switches.

- Beginning with Cisco NX-OS Release 10.1(2), 802.1X is supported on Cisco Nexus 9300-GX platform switches.

- Beginning with Cisco NX-OS Release 10.2(1)F, 802.1X is supported on the N9K-C9364D-GX2A and N9K-C9332D-GX2B switches.

- Beginning with Cisco NX-OS Release 10.3(2)F, MAC authentication bypass and multi-auth are supported on Cisco Nexus 9508 switches with N9K-X9788TC-FX, and N9K-X97160YC-EX line cards.

- The Cisco Nexus C9508 switches with N9K-X9788TC-FX, and N9K-X97160YC-EX line cards does not support the following features with 802.1X :

    - DVLAN

    - DACL

- FEX-AA

- VXLAN and mac-move

- CoA

- Only MAB supported as authentication method and no EAP

- Support is for access port with single access VLAN.

- The following platform limitation is applicable only for Cisco Nexus 9000 PX/TX/PQ EoR or ToR switches:

  - When feature 802.1X is configured on a vPC domain, the traffic traversing the peer-link may get punted to CPU if the source MAC belongs to the vPC peer and traffic needs to be bridged over the same VLAN to an orphan port.

- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 underlay is supported on 802.1X for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.

- Beginning with Cisco NX-OS Release 10.4(1)F, Cisco Nexus 9336C-FX2, 93180YC-FX3, 93108TC-FX3P switches and Cisco Nexus 9500 switches with X9716D-GX line cards supports 802.1X port-based authentication using EAP/EAP-TLS (to carry certificates) for uplink ports where MACSec is required with the following limitations:

  - EAP-TLS supported TLS version is 1.2.

  - Support for Single EAP profile per switch and multiple interfaces can use the same EAP profile.

  - No support for MAC Move profiles of supplicants.

  - Authenticator profile will be enabled for L3 ports, trunks ports, vPC for only MACsec EAP-TLS.

> **Note** 802.1X authenticator functionality for MAB/EAP clients will not be supported for L3 or Trunk and vPC ports.

  - EAP-TLS is supported for only EAP on MACsec configured interfaces.

  - EAP-TLS is supported only on Multi-Host mode.

  - DACL/Critical AUTH/FEX-AA and other 802.1X features on 802.1X MACsec enabled interfaces is not supported.

  - EAP-TLS is supported for only remote authentication (ISE/RADIUS – ISE 3.0 and above), local authentication is not supported.

  - The following order must be followed for EAP-TLS configuration to function properly:

    - The **macsec eap policy** command must be configured first and then the **dot1x supplicant eap profile TLS** command.

    - For the **no** form of the EAP profile command, the **dot1x supplicant eap profile TLS** command must be removed first and then the **macsec eap policy** command.

- For **no feature** command, We recommend to remove the 802.1X feature first and then MACsec feature to avoid DME DB inconsistencies.

- Single EAP profile which is configured across the switch can be applied on different interface.

- If **macsec eap policy** is configured on interfaces, the regular 802.1X authenticator function or commands are not supported.

- Peer to peer MACsec enabled switches must have same 802.1X or MACsec configurations.

- If the commands are different (like one side should-secure and another side must-secure), the behavior will be undefined and must trigger shut/no-shut to recover.

- Once MACsec secure session is created with a trust point and eap profile is added to interface:

    - Removal of trustpoint configuration will not delete MACsec session.

    - Removal of 802.1X supplicant command will not delete MACsec session.

    - MACsec session will be deleted only on MACsec interface specific command removal.

- MACsec PKI is supported on switches without any intermediate switches or hops and should be directly connected.

- MACsec PKI (802.1X EAP-TLS) mode does not support EoR Stateful Switch Over (SSO).

- EAP-TLS is supported only on the following interface types:

    - L2/L3 ports, Port-channel member ports, trunk ports and breakout ports

    - Unsupported interface types – there is no command level restriction.

- Number of MACsec sessions supported depends on the physical interface scale.

- Beginning with Cisco NX-OS Release 10.4(3)F, EAP-TLS supports Transport Layer Security version 1.3 and 1.2 on Cisco Nexus switches.

> **Note**    If the RADIUS server is not capable of TLS v1.3, then TLS v1.2 is used, as it is the minimum supported version.

# 802.1X Guidelines and Limitations for Voice VLAN

- Beginning with Cisco NX-OS Release 10.5(2)F, the 802.1X Voice VLAN feature is supported on the following Cisco Nexus switches:

    - N9K-C9348GC-FXP

    - N9K-C93108TC-FX3P

    - N9K-C9348GC-FX3

    - N9K-C9348GC-FX3PH

- Only L2 switchport mode access is supported with voice VLAN configuration.

- Only one voice VLAN and one data VLAN are supported.

- Dynamic Access Control Lists (DACL) and Dynamic VLAN (DVLAN) is not supported for Multi-Domain Authentication (MDA) ports.

- The following commands are not supported for voice VLAN configuration on ports configured with 802.1X:

  - **switchport voice vlan untagged**

  - **switchport voice vlan dot1p**

  If these configurations are present on the 802.1X port, the phone will not be authorized. Therefore, the multi-domain port must be configured with a specific voice VLAN ID.

- Having the same value for both access VLAN and voice VLAN is not supported. If configured with the same value, ensure you change the value and flap the port.

# Guidelines and Limitations for Per-User DACL Support for 802.1X

- The following switch platforms support this feature:

  - Cisco Nexus 9300-EX platform switches

  - Cisco Nexus 9300-FX platform switches

  - Cisco Nexus 9300-FX2 platform switches

- Per-user DACL supports the IPv4 TCP, UDP, and ICMP ACL rules, but doesn't support IPv6 ACL rules.

- Per-user DACLs are limited to single RADIUS response which is less than 4KB and maximum number of ACEs supported is 32.

- This feature doesn't support standard ACLs on the switch port.

- Only one DACL per port is supported. The maximum number of DACLs supported across a switch is same as the number of ports in that switch.

- DACL and dynamic VLAN aren't supported together on the same port.

- Dynamically modifying DACL content from ISE is not supported. To achieve this, clear the previously applied DACL from the port using the **clear dot1x interface** command and then the new one from ISE is applied. With that, all the clients on this port will have transient traffic disruption.

- Cisco Nexus 9000 series switches in AA FEX mode do not support the per-user DACL.

- Per-user DACL supports only MAB and multi-auth host mode.

- Like all other Nexus 9000 802.1x features, per-User DACL is also supported only on physical ports, that is, regular L2 access ports and not supported on trunk, vPC, port-channel and its members, and subinterfaces.

- Like all other Nexus 9000 ACLs applied on the switch, the maximum limit of the per-user DACL is 4000 ASCII characters.

- MAC-move profiles for the per user DACL feature isn't supported.

- Beginning with Cisco NX-OS Release 10.2(1), the DACL feature is supported on Cisco Nexus 9300-FX/FX2/EX TOR switches.

- Beginning with Cisco NX-OS Release 10.5(2)F, the DACL feature is supported on Cisco Nexus 9300-FX3,GX,GX2,H2R, and H1 Series switches.

# Guidelines and Limitations for Critical Authentication

- Critical authentication supports only for basic MAB clients and not supported on topologies like FEX-AA and VxLAN.

- Enabling the **authentication event server dead action authorize** command all the time is a security risk because all the unauthorized client traffic is allowed.

- Beginning with Cisco NX-OS Release 10.1(2), the critical authentication feature is supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX TOR switches.

- Beginning with Cisco NX-OS Release 10.2(1)F, the critical authentication feature is supported on the N9K-C9364D-GX2A and N9K-C9332D-GX2B switches.

- Beginning with Cisco NX-OS Release 10.3(2)F, the critical authentication feature is supported on Cisco Nexus 9508 switches with N9K-X9788TC-FX, and N9K-X97160YC-EX line cards.

# Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

**Table 14: 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**)<br><br>**Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant. |
| Periodic reauthentication | Disabled |
| Number of seconds between reauthentication attempts | 3600 seconds |
| Quiet timeout period | 60 seconds (number of seconds that the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant) |

| Parameters | Default |
|---|---|
| Retransmission timeout period | 30 seconds (number of seconds that the Cisco NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request) |
| Maximum retransmission number | 2 times (number of times that the Cisco NX-OS device will send an EAP-request/identity frame before restarting the authentication process) |
| Host mode | Single host |
| Supplicant timeout period | 30 seconds (when relaying a request from the authentication server to the supplicant, the amount of time that the Cisco NX-OS device waits for a response before retransmitting the request to the supplicant) |
| Authentication server timeout period | 30 seconds (when relaying a response from the supplicant to the authentication server, the amount of time that the Cisco NX-OS device waits for a reply before retransmitting the response to the server) |

# Configuring 802.1X

This section describes how to configure the 802.1X feature.

# Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

**Procedure**

---

**Step 1**   Enable the 802.1X feature.

**Step 2**   Configure the connection to the remote RADIUS server.

**Step 3**   Enable 802.1X feature on the Ethernet interfaces.

---

# Enabling the 802.1X Feature

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

**Procedure**

|        | **Command or Action**                                    | **Purpose**                                  |
| ------ | -------------------------------------------------------- | -------------------------------------------- |
| Step 1 | **configure terminal**                                   | Enters global configuration mode.            |
|        | **Example:**                                             |                                              |
|        | switch# **configure terminal**<br>switch(config)#        |                                              |
| Step 2 | **feature dot1x**                                        | Enables the 802.1X feature. The default is   |
|        | **Example:**                                             | disabled.                                    |
|        | switch(config)# **feature dot1x**                        |                                              |
| Step 3 | **exit**                                                 | Exits configuration mode.                    |
|        | **Example:**                                             |                                              |
|        | switch(config)# **exit**<br>switch#                      |                                              |
| Step 4 | (Optional) **show dot1x**                                | Displays the 802.1X feature status.          |
|        | **Example:**                                             |                                              |
|        | switch# **show dot1x**                                   |                                              |
| Step 5 | (Optional) **copy running-config startup-config**        | Copies the running configuration to the startup |
|        | **Example:**                                             | configuration.                               |
|        | switch# **copy running-config**<br>**startup-config**    |                                              |

# 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**                                      | Enters global configuration mode.             |
|        | **Example:**                                                |                                               |
|        | switch# configure terminal<br>switch(config)#               |                                               |
| Step 2 | **aaa authentication dot1x default group** *group-list*     | Specifies the RADIUS server groups to use for |
|        |                                                             | 802.1X authentication.                        |
|        | **Example:**                                                |                                               |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# aaa authentication dot1x default group rad2` | The *group-list* argument consists of a space-delimited list of group names. The group names are the following:<br><br>• **radius**—Uses the global pool of RADIUS servers for authentication.<br><br>• *named-group* —Uses the global pool of RADIUS servers for authentication. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show radius-server**<br><br>**Example:**<br><br>`switch# show radius-server` | Displays the RADIUS server configuration. |
| **Step 5** | (Optional) **show radius-server group** [*group-name*]<br><br>**Example:**<br><br>`switch# show radius-server group rad2` | Displays the RADIUS server group configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`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**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot* / *port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | **dot1x port-control** {**auto** \| **force-authorized** \| **forced-unauthorized**}<br><br>**Example:**<br><br>`switch(config-if)# dot1x port-control`<br>`auto` | Changes the 802.1X authentication state on the interface. The default is force-authorized. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 5** | (Optional) **show dot1x all**<br><br>**Example:**<br><br>`switch# show dot1x all` | Displays all 802.1X feature status and configuration information. |
| **Step 6** | (Optional) **show dot1x interface ethernet** *slot* / *port*<br><br>**Example:**<br><br>`switch# show dot1x interface ethernet`<br>`2/1` | Displays 802.1X feature status and configuration information for an interface. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

## Configuring 802.1X for Voice VLAN

Beginning with Cisco NX-OS Release 10.5(2)F, you can enable multi-domain 802.1X authentication on a single port.

**Before you begin**

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot* / *port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | **dot1x host-mode multi-domain**<br><br>**Example:**<br><br>`switch(config-if)# dot1x host-mode`<br>`multi-domain` | Enables or disables the multi-domain host mode at the interface level.<br><br>Use **no** form of this command to disable the multi-domain host mode at the interface level.<br><br>**Note**    Voice clients will be successfully authenticated only on ports with host-mode multi-domain. If the host-mode is anything else, the voice client will fail authentication and be disabled. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

# Configuring EAP-TLS

Beginning with Cisco NX-OS Release 10.4(1)F, you can use EAP-TLS profile for 802.1X authentication.

**Before you begin**

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

- On the interface, configure the MACsec EAP policy and then attach the **dot1x supplicant eap profile**. For configuring MACsec EAP policy, see Configuring MACsec EAP , on page 623 section.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **[no] eap profile TLS**<br><br>**Example:**<br><br>`switch(config)# eap profile TLS`<br>`switch(config-eap-profile)#` | Configures the 802.1X EAP profile mode.<br><br>The **no** form of the command is used to remove the eap profile. |
| Step 3 | **pki-trustpoint** *trustpoint name*<br><br>**Example:**<br><br>`switch(config-eap-profile)#`<br>`pki-trustpoint tp1`<br>`switch(config-eap-profile)#` | Specifies the trustpoint to be used. |
| Step 4 | **method** *type*<br><br>**Example:**<br><br>`switch(config-eap-profile)# method TLS`<br>`switch(config-eap-profile)#` | Enters global configuration mode.<br><br>Specifies the EAP method to be used. |
| Step 5 | **interface ethernet** *slot / port*<br><br>**Example:**<br><br>`switch(config-eap-profile)# interface`<br>`ethernet 1/30`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| Step 6 | **[no] dot1x supplicant eap profile** *eap profile name*<br><br>**Example:**<br><br>`switch(config-if)# dot1x supplicant eap`<br>` profile` | Enters global configuration mode.<br><br>Configures the 802.1X supplicant to the EAP profile. |

# 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**<br><br>**Example:** | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# configure terminal`<br>`switch(config)#` | |
| Step 2 | (Optional) **show dot1x interface ethernet** *slot*/*port*<br><br>**Example:**<br>`switch# show do1x interface ethernet 2/1` | Displays the 802.1X configuration on the interface. |
| Step 3 | **interface ethernet** *slot*/*port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| Step 4 | [**no**] **dot1x pae authenticator**<br><br>**Example:**<br>`switch(config-if)# dot1x pae`<br>`authenticator` | Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface.<br><br>**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**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Enabling Critical Authentication

### Before you begin

- Enable monitoring of RADIUS.

- Ensure that all servers in the group are RADIUS servers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **radius-server test idle-time** *minutes*<br><br>**Example:** | Specifies parameters for global server monitoring. The default username is test, and the default password is test. The default value |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# **radius-server test idle-time 1** | for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes. |
| | | **Note** For periodic RADIUS server monitoring, the idle timer value must be greater than 0. If there are multiple servers in the group, set the idle timer to 1 for each server. |
| Step 3 | **radius-server deadtime** *minutes*<br><br>**Example:**<br><br>switch(config)# **radius-server deadtime 1** | Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.<br><br>**Note** Set the dead time to a value greater than 0 to enable monitoring. |
| Step 4 | **radius-server host** *ipv4-address* **key**[**0** \|**6** \|**7**] *key-value*<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.105.222.183 key 7 "fewhg" authentication accounting** | Specifies a RADIUS key for all RADIUS servers. You can specify if the key-value is in clear text format (**0**), type-6 encrypted (**6**), or type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no RADIUS key is configured.<br><br>**Note** If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configuring the Shared Secret for RADIUS or TACACS+. |
| Step 5 | **radius-server host** *ipv4-address* **test idle-time** *minutes*<br><br>**Example:**<br><br>switch(config)# **radius-server host 10.105.222.183 test idle-time 1** | Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.<br><br>**Note** For periodic RADIUS server monitoring, set the idle timer to a value greater than 0. |
| Step 6 | **aaa group server radius** *group-name*<br><br>**Example:**<br><br>switch(config)# **aaa group server radius ISE_2.4**<br>switch(config-radius)# | Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | To delete a RADIUS server group, use the **no** form of this command. |
| | | **Note** You are not allowed to delete the default system-generated default group (RADIUS). |
| **Step 7** | **server** {*ipv4-address* \| *ipv6-address* \| *hostname*}<br><br>**Example:**<br>`switch(config-radius)# server 10.105.222.183` | Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command. |
| **Step 8** | **use-vrf** *vrf-name*<br><br>**Example:**<br>`switch(config-radius)# use-vrf management` | Specifies the VRF to use to contact the servers in the server group. |
| **Step 9** | **source-interface** *interface*<br><br>**Example:**<br>`switch(config-radius)# source-interface mgmt 0` | Configures the global source interface for all RADIUS server groups configured on the device. |
| **Step 10** | **exit**<br><br>**Example:**<br>`switch(config-radius)# exit`<br>`switch(config)#` | Exits the RADIUS server group configuration submode. |
| **Step 11** | **authentication event server dead action authorize**<br><br>**Example:**<br>`switch(config)# authentication event server dead action authorize` | Authorizes all the clients when the RADIUS server is unreachable. |

# 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**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface ethernet** *slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| Step 3 | **dot1x re-authentication**<br>**Example:**<br>`switch(config-if)# dot1x`<br>`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*<br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`re-authperiod 3300` | Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535.<br><br>**Note**    This command affects the behavior of the Cisco NX-OS device only if you enable periodic reauthentication on the interface. |
| Step 5 | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits configuration mode. |
| Step 6 | (Optional) **show dot1x all**<br>**Example:**<br>`switch(config)# show dot1x all` | Displays all 802.1X feature status and configuration information. |
| Step 7 | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config`<br>`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*] | Reauthenticates the supplicants on the Cisco |
|        | **Example:**                                       | NX-OS device or on an interface.         |
|        | ```                                                |                                          |
|        | switch# dot1x re-authenticate interface            |                                          |
|        | 2/1                                                |                                          |
|        | ```                                                |                                          |

# Changing 802.1X Authentication Timers for an Interface

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

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

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

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

**Switch-to-supplicant retransmission timer for EAP response frames**
The supplicant responds to the EAP-request/identity frame from the Cisco NX-OS device with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

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

**Inactive period timeout**
When the Cisco NX-OS device remains inactive for a set period of time. The timeout inactivity-period value determines the inactive period. The recommended minimum value is 1800 seconds. You must ensure that the value is less than the value of the re-authentication time.

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

**Before you begin**

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot*/*port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | (Optional) **dot1x timeout quiet-period** *seconds*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`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*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`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*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`server-timeout 60` | Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds. |
| **Step 6** | (Optional) **dot1x timeout supp-timeout** *seconds*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`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*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout`<br>`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. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 8** | (Optional) **dot1x timeout inactivity-period** *seconds*<br><br>**Example:**<br>`switch(config-if)# dot1x timeout inactivity-period 1800` | Sets the number of seconds the switch can remain inactive. The recommended minimum value is1800 seconds. |
| **Step 9** | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 10** | (Optional) **show dot1x all**<br><br>**Example:**<br>`switch# show dot1x all` | Displays the 802.1X configuration. |
| **Step 11** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Enabling MAC Authentication Bypass

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

**Before you begin**

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

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot*/*port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | **dot1x mac-auth-bypass** [**eap**]<br><br>**Example:**<br>`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. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config-if)# exit`<br>`switch(config)#` | Exits configuration mode. |
| **Step 5** | (Optional) **show dot1x all**<br><br>**Example:**<br><br>`switch# show dot1x all` | Displays all 802.1X feature status and configuration information. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring the Default 802.1X Authentication Method - MAB

Beginning with Cisco NX-OS Release 9.3(5), all traffic that is received on the 802.1X enabled ports can be authenticated only by MAC authentication bypass (MAB). Prior to Cisco NX-OS Release 9.3(5), all traffic was first authenticated by EAPOL and authentication by MAB occurred only after the EAPOL authentication session timed out.

**Before you begin**

Enable the MAB feature on the Cisco NX-OS device.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot/port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)` | Selects the interface and enters interface configuration mode. |
| **Step 3** | **dot1x mac-auth-bypass**<br><br>**Example:**<br><br>`switch(config-if)# dot1x mac-auth-bypass` | Enables MAC authentication bypass. The default is bypass disabled. |
| **Step 4** | [**no**]**dot1x authentication order mab**<br><br>**Example:**<br><br>`switch(config-if)# dot1x authentication order mab` | Enables MAB for the authentication of the data traffic with the radius server. The no form of this command changes the default authentication method to EAPOL. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **exit**<br><br>**Example:**<br><br>`switch(config-if)# exit`<br>`switch(config)#` | Exits configuration mode. |
| **Step 6** | (Optional) **show dot1x all**<br><br>**Example:**<br><br>`switch# show dot1x all` | Displays the 802.1X feature status and configuration information. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Creating Dynamic Access Lists

### Before you begin

Ensure the following:

- Pre-program the ACL name (acl-name) with all the ACEs to allow or block specific traffic class for the 802.1X MAB client. The configured ACL name (acl-name) on the device must match the acl-name received from the ISE Server.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **hardware access-list tcam region ing-dacl** *tcam size*<br><br>**Example:**<br><br>`switch(config)# hardware access-list tcam`<br>`region ing-dacl 256`<br>`switch(config)#` | Specifies the TCAM size. The range is between 0 to 2147483647. |
| **Step 3** | **ip access-list** *blacklist*<br><br>**Example:**<br><br>`switch(config)# ip access-list`<br>`creative_blacklist` | Configures the defined blacklist and applies it based on the configured TCAM size. |

| | Command or Action | Purpose |
|---|---|---|
| Step 4 | (Optional) **show ip access-list**<br><br>**Example:**<br>`switch(config)# ip access-list`<br>`creative_blacklist1` | Displays the configured IP access list. |
| Step 5 | (Optional) **show ip access-list dynamic**<br><br>**Example:**<br>`switch(config)# ip access-list`<br>`creative_blacklist1_new_Ethernet1/1`<br>`statistics per-entry`<br>`10 permit udp 0000.1b40.ff13`<br>`0000.0000.0000 any range bootps bootpc`<br>`vlan 100 [match=123]`<br>`20 permit udp 0000.1b40.ff13`<br>`0000.0000.0000 any eq domain vlan 100`<br>`[match=456]`<br>`30 deny 0000.1b40.ff13 0000.0000.0000`<br>`any [match=789]` | Displays the configured IP access list. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring Per-User DACLs

You can configure per-user DACLs in the Cisco ISE server. You can then implement it in your authorization policies for control of how different users and groups of users access the network.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **hardware access-list tcam region ing-dacl**<br><br>**Example:**<br>`switch(config)# hardware access-list tcam`<br>`region ing-dacl` | Configures TCAM on the switch to create a new DACL-TCAM region. |
| Step 3 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 4 | **reload**<br><br>**Example:**<br>`switch# reload` | Reloads the Cisco NX-OS device. |

**What to do next**

Configure the DACL for the blocklisted clients on ISE.

**Note** The ACEs on ISE shouldn't have a deny rule for IP because an implicit deny is internally added for every DACL client.

The blocklist client connects to the 802.1X port and downloads the ACL AV-Pair as part of the radius access-accept message. The received ACL is then applied on the port for the particular client.

For more information about how to configure the DACLs, see the *Configure Permissions for Downloadable ACLs* section in the *Segmentation* chapter of the *Cisco Identity Services Engine Administrator Guide, Release 3.0*.

# 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**<br><br>**Example:**<br>`switch# `**`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface ethernet** *slot/port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)` | Selects the interface to configure and enters interface configuration mode. |
| Step 3 | **dot1x host-mode** {**multi-host** \| **single-host**}<br><br>**Example:**<br>`switch(config-if)# `**`dot1x host-mode`**<br>**`multi-host`** | Configures the host mode. The default is single-host.<br><br>**Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **dot1x host-mode multi-auth**<br><br>**Example:**<br>switch(config-if)# **dot1x host-mode multi-auth** | Configures the multiple authentication mode. The port is authorized only on a successful authentication of either EAP or MAB or a combination of both. Failure to authenticate will restrict network access.<br><br>authentication either EAP or MAB |
| **Step 5** | **exit**<br><br>**Example:**<br>switch(config-if)# **exit**<br>switch(config)# | Exits configuration mode. |
| **Step 6** | (Optional) **show dot1x all**<br><br>**Example:**<br>switch# **show dot1x all** | Displays all 802.1X feature status and configuration information. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# **copy running-config startup-config** | Copies the running configuration to the startup configuration. |

# 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**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **no dot1x system-auth-control**<br><br>**Example:**<br><br>`switch(config)# no dot1x system-auth-control` | Disables 802.1X authentication on the Cisco NX-OS device. The default is enabled.<br><br>**Note**   Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Cisco NX-OS device. |
| Step 3 | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| Step 4 | (Optional) **show dot1x**<br><br>**Example:**<br><br>`switch# show dot1x` | Displays the 802.1X feature status. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`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**<br><br>**Example:**<br><br>`switch# ` **`configure terminal`**<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **no feature dot1x**<br><br>**Example:**<br><br>`switch(config)# ` **`no feature dot1x`** | Disables 802.1X.<br><br>**Caution**   Disabling the 802.1X feature removes all 802.1X configuration. |

| | Command or Action | Purpose |
|---|---|---|
| Step 3 | **exit**<br><br>**Example:**<br>`switch(config)# `**`exit`**<br>`switch#` | Exits configuration mode. |
| Step 4 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# `**`copy running-config`**<br>**`startup-config`** | Copies the running configuration to the startup configuration. |

# Resetting the 802.1X Interface Configuration to the Default Values

You can reset the 802.1X configuration for an interface to the default values.

### Before you begin

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface ethernet** *slot*/*port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)` | Selects the interface to configure and enters interface configuration mode. |
| Step 3 | **dot1x default**<br><br>**Example:**<br>`switch(config-if)# dot1x default` | Reverts to the 802.1X configuration default values for the interface. |
| Step 4 | **exit**<br><br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits configuration mode. |
| Step 5 | (Optional) **show dot1x all**<br><br>**Example:**<br>`switch(config)# show dot1x all` | Displays all 802.1X feature status and configuration information. |

|        | Command or Action | Purpose |
|--------|-------------------|---------|
| **Step 6** | (Optional) **copy running-config startup-config** <br> **Example:** <br> `switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Setting the Maximum Authenticator-to-Supplicant Frame for an Interface

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

### Before you begin

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

### Procedure

|        | Command or Action | Purpose |
|--------|-------------------|---------|
| **Step 1** | **configure terminal** <br> **Example:** <br> `switch# configure terminal` <br> `switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot*/*port* <br> **Example:** <br> `switch(config)# interface ethernet 2/1` <br> `switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | **dot1x max-req** *count* <br> **Example:** <br> `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. <br> **Note**    Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface. |
| **Step 4** | **exit** <br> **Example:** <br> `switch(config)# exit` <br> `switch#` | Exits interface configuration mode. |
| **Step 5** | (Optional) **show dot1x all** <br> **Example:** <br> `switch# show dot1x all` | Displays all 802.1X feature status and configuration information. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`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**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **dot1x radius-accounting**<br><br>**Example:**<br><br>`switch(config)# dot1x radius-accounting` | Enables RADIUS accounting for 802.1X. The default is disabled. |
| **Step 3** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show dot1x**<br><br>**Example:**<br><br>`switch# show dot1x` | Displays the 802.1X configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`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**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface ethernet** *slot/port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| Step 3 | **dot1x max-reauth-req** *retry-count*<br><br>**Example:**<br>`switch(config-if)# dot1x max-reauth-req`<br>` 3` | Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10. |
| Step 4 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits interface configuration mode. |
| Step 5 | (Optional) **show dot1x all**<br><br>**Example:**<br>`switch# show dot1x all` | Displays all 802.1X feature status and configuration information. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Verifying the 802.1X Configuration

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

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

| Command | Purpose |
|---|---|
| **show startup-config dot1x** | Displays the 802.1X feature configuration in the startup configuration. |

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

The following example displays information about the EAP-TLS configuration on the port as both authenticator and supplicant in authorized state:

```
switch(config)# show dot1x int eth 5/6 details

Dot1x Info for Ethernet5/6
----------------------------------
                      PAE = AUTHENTICATOR
              PortControl = AUTO
                 HostMode = MULTI HOST
        ReAuthentication = Disabled
              QuietPeriod = 60
            ServerTimeout = 30
              SuppTimeout = 30
             ReAuthPeriod = 3600 (Locally configured)
               ReAuthMax = 2
                  MaxReq = 2
                 TxPeriod = 30
          RateLimitPeriod = 0
         InactivityPeriod = 0
          Mac-Auth-Bypass = Disabled

Dot1x Info for Ethernet5/6
----------------------------------
                      PAE = SUPPLICANT
              StartPeriod = 30
               AuthPeriod = 30
               HeldPeriod = 60
                 MaxStart = 3

Dot1x Authenticator Client List
-------------------------------
                Supplicant = C4:B2:39:2C:EE:50
                    Domain = DATA
              Auth SM State = AUTHENTICATED
         Auth BEND SM State = IDLE
               Port Status = AUTHORIZED
       Authentication Method = EAP
           Authenticated By = Remote Server
                  Auth-Vlan = 0
               DACL-Applied = False

Dot1x Supplicant Client List
-------------------------------
              Authenticator = C4:B2:39:2C:EE:50
             Supp SM State = AUTHENTICATED
          Supp Bend SM State = IDLE
                Port Status = AUTHORIZED
```

# 802.1X Support for VXLAN EVPN

This section describes how to configure 802.1X for VXLAN EVPN.

# Guidelines and Limitations for 802.1X Support for VXLAN EVPN

The following are the guidelines and limitations for 802.1X support for VXLAN EVPN:

- Beginning with Cisco NX-OS Release 9.3(7), 802.1X support for VXLAN EVPN feature is supported for Cisco Nexus 9300-GX platform switches.

- Port channel interfaces or the member ports of the port channel are not supported.

- vPC ports are not supported.

- The current support of the feature uses regular and dynamic EVPN updates on the BGP-EVPN control plane for 802.1X secure MAC updates. As a result, we cannot prevent the move across EVPN even if the global policy is "dot1x mac-move deny".

- Ensure that the "dot1x mac-move" policy is configured the same across the fabric. There is no configuration validation across the nodes, hence it could lead to unexpected behavior if the configuration policy is not in sync.

- The local to remote MAC moves behavior for the deny and permit modes is permitted. Therefore, the MAC move is permitted even if the deny mode is enabled.

- Ensure that the 802.1X and the port-security ports use different VLANs. The same VLAN cannot be assigned to both ports.

- 802.1X is not VLAN aware and hence having the same MAC in two different VLANs is not possible. Depending on the mac-move mode that is selected, either the MAC is moved to a new VLAN or it is denied.

- You cannot configure static and secure MAC together.

- Cisco Nexus 9504 and Cisco Nexus 9508 platform switches with -R line cards does not support multi-authentication and multi-authentication with VXLAN.

- RADIUS change of Authorization is supported for VXLAN EVPN.

- The recommended re-authentication time interval for a scale setup is the default value, which is 3600 seconds.

- 802.1X is not supported with Fabric Peering

# Configuring 802.1X Support for VXLAN EVPN

This procedure configures 802.1X for VXLAN EVPN.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# `**`configure terminal`** | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **feature dot1x**<br><br>**Example:**<br><br>switch(config)# **feature dot1x** | Enables the 802.1X feature. The default is disabled. |
| Step 3 | **dot1x mac-move {permit \| deny}**<br><br>**Example:**<br><br>switch(config)# **dot1x mac-move permit** | The **deny** parameters denies MAC moves. The **permit** parameter permits MAC moves. |
| Step 4 | (Optional) **show running-config dot1x all**<br><br>**Example:**<br><br>switch(config)# **show running-config dot1x all**<br><br>!Command: show running-config dot1x all<br>!No configuration change since last restart<br>!Time: Thu Sep 20 10:22:58 2018<br><br>version 9.2(2) Bios:version 07.64<br>feature dot1x<br><br>dot1x system-auth-control<br>dot1x mac-move deny<br><br><br>interface Ethernet1/1<br>  dot1x host-mode multi-auth<br>  dot1x pae authenticator<br>  dot1x port-control auto<br>  no dot1x re-authentication<br>  dot1x max-req 1<br>  dot1x max-reauth-req 2<br>  dot1x timeout quiet-period 60<br>  dot1x timeout re-authperiod 3600<br>  dot1x timeout tx-period 1<br>  dot1x timeout server-timeout 30<br>  dot1x timeout ratelimit-period 0<br>  dot1x timeout supp-timeout 30<br>  dot1x timeout inactivity-period 0<br>  dot1x mac-auth-bypass<br><br>interface Ethernet1/33<br>  dot1x host-mode multi-auth<br>  dot1x pae authenticator<br>  dot1x port-control auto<br>  no dot1x re-authentication<br>  dot1x max-req 1<br>  dot1x max-reauth-req 2<br>  dot1x timeout quiet-period 60<br>  dot1x timeout re-authperiod 3600<br>  dot1x timeout tx-period 1<br>  dot1x timeout server-timeout 30<br>  dot1x timeout ratelimit-period 0<br>  dot1x timeout supp-timeout 30<br>  dot1x timeout inactivity-period 0<br>  dot1x mac-auth-bypass | Displays the 802.1X configuration. |

# Verifying the 802.1X Support for VXLAN EVPN

To display the 802.1X support for VXLAN EVPN configuration information, enter one of the following commands:

| Command | Purpose |
|---|---|
| **show running-config dot1x all** | Displays 802.1X running configuration. |
| **show dot1x all summary** | Displays the interface status. |
| **show dot1x** | Displays the default settings. |
| **show dot1x all** | Displays additional interface detail. |

**Example of show running-config dot1x all command**

```
switch# show running-config dot1x all
!Command: show running-config dot1x all
!No configuration change since last restart
!Time: Thu Sep 20 10:22:58 2018

version 9.2(2) Bios:version 07.64
feature dot1x

dot1x system-auth-control
dot1x mac-move deny


interface Ethernet1/1
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
  dot1x max-req 1
  dot1x max-reauth-req 2
  dot1x timeout quiet-period 60
  dot1x timeout re-authperiod 3600
  dot1x timeout tx-period 1
  dot1x timeout server-timeout 30
  dot1x timeout ratelimit-period 0
  dot1x timeout supp-timeout 30
  dot1x timeout inactivity-period 0
  dot1x mac-auth-bypass

interface Ethernet1/33
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
  dot1x max-req 1
  dot1x max-reauth-req 2
  dot1x timeout quiet-period 60
  dot1x timeout re-authperiod 3600
  dot1x timeout tx-period 1
  dot1x timeout server-timeout 30
  dot1x timeout ratelimit-period 0
  dot1x timeout supp-timeout 30
  dot1x timeout inactivity-period 0
  dot1x mac-auth-bypass
```

## Example of the show dot1x all summary command

```
switch# show dot1x all summary

          Interface    PAE           Client        Status
----------------------------------------------------------------
          Ethernet1/1  AUTH          none    UNAUTHORIZED

          Interface    PAE           Client        Status
----------------------------------------------------------------
          Ethernet1/33 AUTH  00:16:5A:4C:00:07    AUTHORIZED
                             00:16:5A:4C:00:06    AUTHORIZED
                             00:16:5A:4C:00:05    AUTHORIZED
                             00:16:5A:4C:00:04    AUTHORIZED
switch#
switch# show mac address-table vlan 10
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
   VLAN     MAC Address       Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10    0016.5a4c.0004    secure    -         T      F    Eth1/33
*   10    0016.5a4c.0005    secure    -         T      F    Eth1/33
*   10    0016.5a4c.0006    secure    -         T      F    Eth1/33
*   10    0016.5a4c.0007    secure    -         T      F    Eth1/33

switch#
switch# show mac address-table vlan 10 (VPC-PEER)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
   VLAN     MAC Address       Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10    0016.5a4c.0004    secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0005    secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0006    secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0007    secure    -         T      F    vPC Peer-Link

switch#
switch# show mac address-table vlan 10 (RVTEP)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
   VLAN     MAC Address       Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C   10    0016.5a4c.0004    dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0005    dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0006    dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0007    dynamic  0         F      F    nve1(67.67.67.67)
```

## Example of the show dot1x command

```
switch# show dot1x
        Sysauthcontrol Enabled
  Dot1x Protocol Version 2
           Mac-Move Deny
```

**Example of the show dot1x all command**

```
switch# show dot1x  all
            Sysauthcontrol Enabled
   Dot1x Protocol Version 2
                  Mac-Move Deny

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

Dot1x Info for Ethernet1/33
----------------------------------
                       PAE = AUTHENTICATOR
               PortControl = AUTO
                  HostMode = MULTI AUTH
         ReAuthentication = Disabled
               QuietPeriod = 60
             ServerTimeout = 30
               SuppTimeout = 30
              ReAuthPeriod = 3600 (Locally configured)
                 ReAuthMax = 2
                    MaxReq = 1
                  TxPeriod = 1
           RateLimitPeriod = 0
          InactivityPeriod = 0
           Mac-Auth-Bypass = Enabled
```

# Verifying Critical Authentication

The following example shows how to view if the critical authentication feature is enabled.

```
switch(config)# show dot1x
            Sysauthcontrol Enabled
      Dot1x Protocol Version 2
                  Mac-Move Permit
  Server-Dead-Action-Authorize Enabled
```

If the value of the **Server-Dead-Action-Authorize** parameter is **Enabled**, the critical authentication feature is enabled.

# Monitoring 802.1X

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

**Before you begin**

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **show dot1x** {**all** | **interface ethernet** *slot*/*port*} **statistics**<br><br>**Example:**<br>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.

# Configuration Example for Per-User DACL

The following example shows the per-user DACL configured on one of the ports. When the DACL is applied, the blocklist traffic is filtered out. If the value of the DACL-Applied parameter is true, the client is a blocklist client, which has received an ACL from ISE.

```
switch# show dot1x all summary
Interface      PAE        Client              Status
Ethernet1/1    AUTH       36:12:61:51:21:52   AUTHORIZED
                          36:12:61:51:21:53   AUTHORIZED

switch# show dot1x all details
---------------------------

Supplicant = 36:12:61:51:21:52
Domain = DATA
```

```
Auth SM State = AUTHENTICATED
DACL-Applied = False
-----------------------------
Supplicant = 36:12:61:51:21:53
Domain = DATA
Auth SM State = AUTHENTICATED
DACL-Applied = True
```

The following example shows how to view the blocklisted traffic.

```
switch# show ip access-list dynamic
IP access list DOT1X_Restricted_base_acl_Ethernet1/1_new statistics per-entry fragments
deny-all
10 permit udp any 3612.6151.2153 0000.0000.0000 any eq 5555 vlan 100 [match=0]
20 permit udp any 3612.6151.2153 0000.0000.0000 any eq 6666 vlan 100 [match=0]
30 deny ip any 3612.6151.2153 0000.0000.0000 any vlan 100 [match=0]
```

# 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* |

**CHAPTER 12**

# Configuring IP ACLs

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

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

This chapter includes the following sections:

## About ACLs

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

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you 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.

# ACL Types and Applications

The device supports the following types of ACLs for security traffic filtering:

**IPv4 ACLs**
> The device applies IPv4 ACLs only to IPv4 traffic.

**IPv6 ACLs**
> The device applies IPv6 ACLs only to IPv6 traffic.

**MAC ACLs**
> The device applies MAC ACLs only to non-IP traffic.

IP and MAC ACLs have the following types of applications:

**Port ACL**
> Filters Layer 2 traffic

**MAC ACL with UDF-based match**
> Filters MAC ACLs with UDF-based match

**Router ACL**
> Filters Layer 3 traffic

**VLAN ACL**
> Filters VLAN traffic

**VTY ACL**
> Filters virtual teletype (VTY) traffic

This table summarizes the applications for security ACLs.

**Table 15: Security ACL Applications**

| Application | Supported Interfaces | Types of ACLs Supported |
|---|---|---|
| Port ACL | • Layer 2 interfaces<br><br>• Layer 2 Ethernet port-channel interfaces<br><br>When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port. | • IPv4 ACLs<br><br>• IPv4 ACLs with UDF-based match<br><br>• IPv6 ACLs<br><br>• IPv6 ACLs with UDF-based match<br><br>• MAC ACLs<br><br>• MAC ACLs with UDF-based match |
| Router ACL | • VLAN interfaces<br><br>• Physical Layer 3 interfaces<br><br>• Layer 3 Ethernet subinterfaces<br><br>• Layer 3 Ethernet port-channel interfaces<br><br>• Management interfaces<br><br>**Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. | • IPv4 ACLs<br><br>• IPv6 ACLs<br><br>**Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.<br><br>**Note** Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports. |

| Application | Supported Interfaces | Types of ACLs Supported |
|---|---|---|
| VLAN ACL | • VLANs | • IPv4 ACLs<br>• IPv6 ACLs<br>• MAC ACLs |
| VTY ACL | • VTYs | • IPv4 ACLs<br>• IPv6 ACLs |

**Related Topics**

# Order of ACL Application

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. Ingress VTY ACL

5. Egress VTY ACL

6. Egress router ACL

7. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs.

**Figure 7: Order of ACL Application**

The following figure shows the order in which the device applies ACLs.

*Figure 8: ACLs and Packet Flow*

The following figure shows where the device applies ACLs, depending upon the type of ACL. The red path indicates a packet sent to a destination on a different interface than its source. The blue path indicates a packet that is bridged within its VLAN.

The device applies only the applicable ACLs. For example, if the ingress port is a Layer 2 port and the traffic is on a VLAN that is a VLAN interface, a port ACL and a router ACL both can apply. In addition, if a VACL is applied to the VLAN, the device applies that ACL too.



# About 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.

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

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

# Protocols for IP ACLs and MAC ACLs

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 or IPv6 ACL, you can specify ICMP by name.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the EtherType number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

In IPv4 and IPv6 ACLs, you can specify protocols by the integer that represents the Internet protocol number.

## 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. How you specify the source and destination depends on whether you are configuring IPv4 ACLs, IPv6 ACLs, or MAC ACLs.

## Implicit Rules for IP and MAC ACLs

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

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

All IPv6 ACLs include the following implicit rule:

```
deny ipv6 any any
```

This implicit rule ensures that the device denies unmatched IPv6 traffic.

**Note**
- IPv6 Neighbor Discovery packets (Router Solicitation, and Router Advertisement) will not be permitted due to the implicit **deny ipv6 any any** rule of an IPv6 ACL.

- You must add the following rules explicitly to allow IPv6 Neighbor Discovery packets in the Cisco Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B, Nexus 9364C and Nexus 9332C platform switches:

  - **permit icmp any any router-advertisement**

  - **permit icmp any any router-solicitation**

- Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages do not match under the implicit rule. The following commands are required to match the NS or NA IPv6 traffic.

  - **permit/deny icmp any any nd-na**

  - **permit/deny icmp any any nd-ns**

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. These options differ by ACL type. The following list includes most but not all additional filtering 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

- Packet length

- IPv6 ACLs support the following additional filtering options:

  - Layer 4 protocol

  - Encapsulating Security Payload

  - Payload Compression Protocol

  - Stream Control Transmission Protocol (SCTP)

  - SCTP, TCP, and UDP ports

  - ICMP types and codes

  - DSCP value

  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set

  - Established TCP connections

  - Packet length

- MAC ACLs support the following additional filtering options:

  - Layer 3 protocol (Ethertype)

  - VLAN ID

  - Class of Service (CoS)

- Beginning Cisco NX-OS Release 9.2(4), IPv4 ACLs and IPv6 in Cisco Nexus 9500 platform switches with N9K-X96136YC-R, N9K-X9636C-R, and N9K-X9636C-RX line cards and N9K-C9504-FM-R fabric module support the following additional filtering options:

  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set

  - Established TCP connections

**Note**

✏️

- TCP flag options are correctly processed by Netstack rather than the kernel (KStack), due to the kernel's lack of support for TCP flags. Additionally, the following syslog message is generated:

  ```
  <HOSTNAME> %NPACL-2-IPT_WARNING: npacl [<#>] WARNING: Mgmt ACL: <ACL>
  Seq:<Seq#> has ACL option: tcp-flags that is not supported in kernel
  stack. Hence that option is not added in its filter rule.
  ```

- The **tcp-flags-mask** option is not supported.

## Sequence Numbers

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

**Adding new rules between existing rules**

By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

**Removing a rule**

Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

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

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

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

**Moving a rule**

With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule 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, Cisco NX-OS 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. Cisco NX-OS supports logical operators in only the ingress direction.

The device stores operator-operand couples in registers called logical operator units (LOUs). The LOU usage for each type of operator is as follows:

**eq**

Is never stored in an LOU

**gt**

Uses 1 LOU

**lt**

Uses 1 LOU

**neq**

Uses 1 LOU

**range**

Uses 1 LOU

> **Note** For range operators, LOU threshold configuration is used to control how the port range is expanded when configuring an ACL entry. If you want to use the LOU operator when the number of the ACL rules exceed the configured threshold value, run the following command: **hardware access-list lou resource threshold** *<x>*, wherein *<x>* denotes the number of ACL rules to be used before the LOU threshold is reached. The range value for *<x>* is 1 to 50, and the default value for LOU threshold is 5.

## ACL Logging

The ACL logging feature monitors ACL flows and logs statistics.

A flow is defined by the source interface, protocol, source IP address, source port, destination IP address, and destination port values. The statistics maintained for a flow include the number of forwarded packets (for each flow that matches the permit conditions of the ACL entry) and dropped packets (for each flow that matches the deny conditions of the ACL entry).

Beginning with Cisco NX-OS Release 10.4(3)F, ACL logging for Security Group ACL (SGACL) is provided on Cisco Nexus 9300-FX3/GX/GX2/H2R platform switches.

For SGACL, a flow is defined by the security group tag (SGT), destination group tag (DGT), source MAC (SMAC), destination MAC (DMAC), SGACL permit/deny information, physical interface on which packet arrived, and hit counts for that particular SGACL flow apart from the basic 5 tuples. To enable the SGACL logging, see Configuring ACL Logging, on page 350.

# Time Ranges

You can use time ranges to control when an ACL rule is in effect. For example, if the device determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the device does not compare the traffic to that rule. The device evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the device updates the affected I/O module whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

IPv4, IPv6, and MAC ACLs support time ranges. When the device applies an ACL to traffic, the rules in effect are as follows:

- All rules without a time range specified

- Rules with a time range that includes the second when the device applies the ACL to traffic

The device supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

**Absolute**

A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:

- Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.

- Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.

- No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.

- No start or end date and time specified—The time range rule is always active.

For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the device automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

**Periodic**

A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The device automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

---

**Note**  The order of rules in a time range does not affect how a device evaluates whether a time range is active. Cisco NX-OS includes sequence numbers in time ranges to make editing the time range easier.

---

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The device determines whether a time range is active as follows:

- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.

- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.

- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

# 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, policy-based routing (PBR), 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.

**Note** Policy-based routing (PBR) ACLs do not support deny access control entries (ACEs) or **deny** commands to configure a rule.

# Kernel Stack ACL

The Kernel Stack ACL is a common CLI infrastructure to configure ACLs for management of inband and outband components.

The Kernel Stack ACL uses NX-OS ACL CLI to secure management applications on management and front panel ports. Configuring a single ACL must be able to secure all management applications on NX-OS.

Kernel Stack ACL is the component that fixes the manual intervention of the user and automatically programs iptable entries when the ACL is applied to mgmt0 interface.

The following is an example for configuring Kernel Stack ACL:

```
swtich# conf t
Enter configuration commands, one per line. End with CNTL/Z.
```

```
switch(config)# ip access-list kacl1
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 deny tcp any any eq 443
switch(config-acl)# 20 permit ip any any
switch(config-acl)# end
switch#

switch(config-if)# interface mgmt0
switch(config-if)#   ip access-group acl1 in
switch(config-if)#   ipv6 traffic-filter acl6 in
switch(config-if)#

switch# sh ip access-lists kacl1
IP access list kacl1
statistics per-entry
10 deny tcp any any eq 443 [match=136]
20 permit ip any any [match=44952]
switch(config)#
```

The following is the Kernel Stack filtering for iptables entries based on the configuration:

```
bash-4.4# ip netns exec management iptables -L -n -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
1 9 576 DROP tcp -- * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:443
2 0 0 ACCEPT all -- * * 0.0.0.0/0 0.0.0.0/0
3 0 0 DROP all -- * * 0.0.0.0/0 0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
bash-4.4#
```

The following are the limitations for the Kernel Stack ACL support:

- This feature is supported only on mgmt0 interface and not on other inband interfaces.

- Five tuples (protocol, source-ip, destination-ip, source-port, and destination-port) of the ACL entry are programmed in the iptables. Rest of the options provided in the ACL entry are not programmed in the iptables and throws a warning syslog in such instances.

  For example, "WARNING: Some ACL options are not supported in kstack. Only partial rule will be installed".

- If the device user has host bash access, then the user can manually update the iptables. This update could potentially corrupt the iptable rules for which they are programmed.

- The verified maximum number of ACEs is 100 for IPv4 traffic and an additional 100 for IPv6 traffic. Throughput may be impacted if more than this scale is applied.

# 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.

**Related Topics**

# Atomic ACL Updates

By default, when a supervisor module of a Cisco Nexus 9000 Series device updates an I/O module with changes to an ACL, it performs an atomic ACL update. An atomic update does not disrupt traffic that the updated ACL applies to; however, an atomic update requires that an I/O module that receives an ACL update has enough available resources to store each updated ACL entry in addition to all pre-existing entries in the affected ACL. After the update occurs, the additional resources used for the update are freed. If the I/O module lacks the required resources, the device generates an error message and the ACL update to the I/O module fails.

If an I/O module lacks the resources required for an atomic update, you can disable atomic updates by using the **no hardware access-list update atomic** command; however, during the brief time required for the device to remove the preexisting ACL and implement the updated ACL, traffic that the ACL applies to is dropped by default.

If you want to permit all traffic that an ACL applies to while it receives a nonatomic update, use the **hardware access-list update default-result permit** command.

This example shows how to disable atomic updates to ACLs:

```
switch# config t
switch(config)# no hardware access-list update atomic
```

This example shows how to permit affected traffic during a nonatomic ACL update:

```
switch# config t
switch(config)# hardware access-list update default-result permit
```

This example shows how to revert to the atomic update method:

```
switch# config t
switch(config)# no hardware access-list update default-result permit
switch(config)# hardware access-list update atomic
```

# Session Manager Support for IP ACLs

Session Manager supports the configuration of IP and MAC ACLs. 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 TCAM Regions

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

On Cisco Nexus 9300 and 9500 Series switches and Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches, the egress TCAM size is 1K, divided into four 256 entries. On Cisco Nexus NFE2-enabled devices (such as the Cisco Nexus 3232C and 3264Q switches), the ingress TCAM size is 6K, divided into twelve 512 slices. Three slices are in one group. On other Cisco Nexus 9300 and 9500 Series switches and the 3164Q and 31128PQ switches, the ingress TCAM size is 4K, divided into eight 256 slices and four 512 slices. A slice is the unit of allocation. A slice can be allocated to one region only. For example, a 512-size slice cannot be used to configure two features of size 256 each. Similarly, a 256-size slice cannot be used to configure two features of size 128 each. The IPv4 TCAM regions are single wide. The IPv6, QoS, MAC, CoPP, and system TCAM regions are double wide and consume double the physical TCAM entries. For example, a logical region size of 256 entries actually consumes 512 physical TCAM entries.

You can create IPv6, port ACLs, VLAN ACLs, and router ACLs, and you can match IPv6 and MAC addresses for QoS. However, Cisco NX-OS cannot support all of them simultaneously. You must remove or reduce the size of the existing TCAM regions (TCAM carving) to enable the IPv6, MAC, or other desired TCAM regions. For every TCAM region configuration command, the system evaluates if the new change can be fit in the TCAM. If not, it reports an error, and the command is rejected. You must remove or reduce the size of existing TCAM regions to make room for new requirements.

On Cisco Nexus 9200 Series switches, the egress TCAM size is 2K, and the ingress TCAM size is 4K. The concepts of TCAM slices and single- and double-wide regions do not apply to these switches. For example, the ing-ifacl region can host IPv4, IPv6, or MAC type entries. IPv4 and MAC types occupy one TCAM entry whereas IPv6 types occupy two TCAM entries.

For N9K-X9636C-RX, when PACL uses external TCAM region, the internal TCAM needs to take 2K for ifacl and the ingress RACL-IPv4 can take upto 2044. Additional four entries are required when egress PACL external TCAM region is used.

ACL TCAM region sizes have the following guidelines and limitations:

- On Cisco Nexus 9500 Series switches, the default ingress TCAM region configuration has one free 256-entry slice in Cisco NX-OS Release 6.1(2)I1(1). This slice is allocated to the SPAN region in Cisco NX-OS Release 6.1(2)I2(1). Similarly, the RACL region is reduced from 2K to 1.5K in Cisco NX-OS Release 6.1(2)I2(1) to make room for the vPC convergence region with 512 entries.

- To enable RACL or PACL on existing TCAM regions, you must carve the TCAM region beyond 12, 288.

- On Cisco Nexus 9300 Series switches, the X9536PQ, X9564PX, and X9564TX line cards are used to enforce the QoS classification policies applied on 40G ports. It has 768 TCAM entries available for carving in 256-entry granularity. These region names are prefixed with "ns-".

- For the X9536PQ, X9564PX, and X9564TX line cards, only the IPv6 TCAM regions consume double-wide entries. The rest of the TCAM regions consume single-wide entries.

- When a VACL region is configured, it is configured with the same size in both the ingress and egress directions. If the region size cannot fit in either direction, the configuration is rejected.

- On Cisco Nexus 9200 Series switches, the ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).

- RACL v6, CoPP, and multicast have default TCAM sizes and these TCAM sizes must be non-zero on the following Cisco Nexus 9504 and Cisco Nexus 9508 line cards to avoid line card failure during reload:

  - N9K-X96136YC-R

  - N9K-X9636C-RX

  - N9K-X9636Q-R

  - N9K-X9636C-R

- When the egress RACL is beyond 4K, the TCAM carving configuration has to be ingress RACL (RACL) + egress RACL (e-racl) summing to 20480. See the following TCAM carving example:

```
hardware access-list tcam region ifacl 0
hardware access-list tcam region ipv6-ifacl 0
hardware access-list tcam region mac-ifacl 0
hardware access-list tcam region racl 0
hardware access-list tcam region ipv6-racl 0
hardware access-list tcam region span 0
hardware access-list tcam region redirect_v4 0
hardware access-list tcam region redirect_v6 0
hardware access-list tcam region e-racl 20480
```

- You can partially use IPv6 RACL with IPv6 IFCAL. This is applicable to Cisco Nexus N9K-C9508 and N9K-C9504 with N9K-X96136YC-R, N9K-X9636C-R, N9K-X9636Q-R, and N9K-X9636C-RX line cards.

- The N9K-X9636C-R and N9K-X9636Q-R line cards support a maximum TCAM region size of 12K. If you configure a greater number, the TCAM region is set to 12K.

- The N9K-X96136YC-R and N9K-X9636C-R line cards support egress RACL of 2K.

- The N9K-X9636C-RX line card supports a TCAM region size beyond 12K. If you configure the RACL IPv4 TCAM region to 100K, the TCAM region is set to 12K for the N9K-X9636C-R and N9K-X9636Q-R line cards and to 100K for the N9K-X9636C-RX line card, provided you have set all of the other TCAM regions and made space for the N9K-X9636C-R and N9K-X9636Q-R line cards to accommodate 12K.

- In addition to the internal TCAM, an external TCAM of 128K is available on the N9K-X9636C-RX line card.

The following table summarizes the regions that need to be configured for a given feature to work. The region sizes should be selected based on the scale requirements of a given feature.

**Table 16: Features per ACL TCAM Region**

| Feature Name | Region Name |
|---|---|
| Port ACL | ifacl: For IPv4 port ACLs |
| | ifacl-udf: For UDFs on IPv4 port ACLs |
| | ing-ifacl: For ingress IPv4, IPv6, and MAC port ACLs |
| | ing-ifacl: For ingress IPv4, IPv6, MAC port ACLs, and MAC port ACLs with UDF |
| | ipv6-ifacl: For IPv6 port ACLs |
| | mac-ifacl: For MAC port ACLs |
| Port QoS (QoS classification policy applied on Layer 2 ports or port channels) | qos, qos-lite, rp-qos, rp-qos-lite, ns-qos, e-qos, or e-qos-lite: For classifying IPv4 packets |
| | ing-l2-qos: For classifying ingress Layer 2 packets |
| | ipv6-qos, rp-ipv6-qos, ns-ipv6-qos, or e-ipv6-qos: For classifying IPv6 packets |
| | mac-qos, rp-mac-qos, ns-mac-qos, or e-mac-qos: For classifying non-IP packets |
| | **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions. |
| VACL | vacl: For IPv4 packets |
| | ipv6-vacl: For IPv6 packets |
| | mac-vacl: For non-IP packets |
| VLAN QoS (QoS classification policy applied on a VLAN) | vqos or ns-vqos: For classifying IPv4 packets |
| | ipv6-vqos or ns-ipv6-vqos: For classifying IPv6 packets |
| | ing-l3-vlan-qos: For classifying ingress Layer 3, VLAN, and SVI QoS packets |
| | mac-vqos or ns-mac-vqos: For classifying non-IP packets |
| | **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions. |

| Feature Name | Region Name |
|---|---|
| RACL | egr-racl: For egress IPv4 and IPv6 RACLs<br><br>e-racl: For egress IPv4 RACLs<br><br>e-ipv6-racl: For egress IPv6 RACLs<br><br>ing-racl: For ingress IPv4 and IPv6 RACLs<br><br>racl: For IPv4 RACLs<br><br>racl-lite: For IPv4 RACLs<br><br>racl-udf: For UDFs on IPv4 RACLs<br><br>ipv6-racl: For IPv6 RACLs |
| Layer 3 QoS (QoS classification policy applied on Layer 3 ports or port channels) | l3qos, l3qos-lite, or ns-l3qos: For classifying IPv4 packets<br><br>ipv6-l3qos or ns-ipv6-l3qos: For classifying IPv6 packets<br><br>**Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve qos regions and the corresponding ns-*qos regions. |
| VLAN source or VLAN filter SPAN (for Cisco Nexus 9500 or 9300 Series switches)<br><br>Rx SPAN on 40G ports (for Cisco Nexus 9300 Series switches only) | span |

| Feature Name | Region Name |
|---|---|
| SPAN filters | ifacl: For filtering IPv4 traffic on Layer 2 (switch port) source interfaces. |
| | ifacl-udf: For UDFs on IPv4 port ACLs |
| | ipv6-ifacl: For filtering IPv6 traffic on Layer 2 (switch port) source interfaces. |
| | mac-ifacl: For filtering Layer 2 traffic on Layer 2 (switch port) source interfaces. |
| | racl-udf: For UDFs on IPv4 RACLs |
| | vacl: For filtering IPv4 traffic on VLAN sources. |
| | ipv6-vacl: For filtering IPv6 traffic on VLAN sources. |
| | mac-vacl: For filtering Layer 2 traffic on VLAN sources. |
| | racl: For filtering IPv4 traffic on Layer 3 interfaces. |
| | ipv6-racl: For filtering IPv6 traffic on Layer 3 interfaces. |
| | ing-l2-span-filter: For filtering ingress Layer 2 SPAN traffic |
| | ing-l3-span-filter: For filtering ingress Layer 3 and VLAN SPAN traffic |
| SVI counters<br><br>**Note** This region enables the packet counters for Layer 3 SVI interfaces. | svi |
| BFD, DHCP relay, or DHCPv6 relay | redirect<br><br>**Note** BFD uses the ing-sup region while DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client use the ing-redirect region. |
| CoPP | copp<br><br>**Note** The region size cannot be 0. |
| System-managed ACLs | system<br><br>**Note** The region size cannot be changed. |

| Feature Name | Region Name |
|---|---|
| vPC convergence | vpc-convergence |
| **Note** This region boosts the convergence times when a vPC link goes down and traffic needs to be redirected to the peer link. | **Note** Setting this region size to 0 might affect the convergence times of vPC link failures. |
| Fabric extender (FEX) | fex-ifacl, fex-ipv6-ifacl, fex-ipv6-qos, fex-mac-ifacl, fex-mac-qos, fex-qos, fex-qos-lite |
| Dynamic ARP inspection (DAI) | arp-ether |
| IP source guard (IPSG) | ipsg |
| Multicast PIM Bidir | mcast_bidir |
| Static MPLS | mpls |
| Network address translation (NAT) | nat |
| NetFlow | ing-netflow |
| OpenFlow | openflow |
| sFlow | sflow |
| Supervisor modules | egr-sup: Egress supervisor<br>ing-sup: Ingress supervisor |

**Related Topics**

# Maximum Label Sizes Supported for ACL Types

Cisco NX-OS switches support the following label sizes for the corresponding ACL types:

**Table 17: ACL Types and Maximum Label Sizes**

| ACL Types | Direction | Label | Label Type |
|---|---|---|---|
| RACL/PBR/VACL/ L3-VLAN QoS/L3-VLAN SPAN ACL | Ingress | 62 | BD |
| PACL/L2 QoS/L2 SPAN ACL | Ingress | 62[1] | IF |
| RACL/VACL/L3-VLAN QoS | Egress | 254 | BD |
| L2 QoS | Egress | 31 | IF |
| RACL | Ingress | 510 | L3 |

[1] The label size can be increased to 62 when you enter the **hardware access-list tcam label ing-ifacl 6** command and reload the switch.

Beginning with Cisco NX-OS Release 9.3(6), the **hardware access-list tcam label ing-ifacl 6** command is introduced and is applicable only for Cisco Nexus 9300-FX platform switches.

Beginning with Cisco NX-OS Release 10.1(2), the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX2 platform switches.

Beginning with Cisco NX-OS Release 10.4(3)F, the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX3, GX, GX2, H2R, H1 platform switches.

# Prerequisites for IP ACLs

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.

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

# Guidelines and Limitations for IP ACLs

IP ACLs have the following configuration guidelines and limitations:

**Note** For more information about the Cisco Nexus 9000 series platform switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- Beginning with Cisco NX-OS Release 10.2(1)F, Egress PACL is supported on the Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches.

- If you configure egress PACL and egress VACL on the same interface, only egress VACL is enabled.

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This recommendation is especially useful for ACLs that include more than 1000 rules. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

- Configuring a IPv4 PACL in the range of 12K to 64K is supported on Cisco Nexus 9500 Series switches with -RX line cards.

- Duplicate ACL entries with different sequence numbers are allowed in the configuration. However, these duplicate entries are not programmed in the hardware access-list.

- Only 62 unique ACLs can be configured. Each ACL takes one label. If the same ACL is configured on multiple interfaces, the same label is shared. If each ACL has unique entries, the ACL labels are not shared, and the label limit is 62. This is not applicable to Cisco Nexus 9500 Series switches and Cisco Nexus 3636C-R switches.

- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:

  - Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.

  - IPv4 packets that have IP options (other IP packet header fields following the destination address field).

  - IPv6 packets that have extended IPv6 header fields.

  Rate limiters prevent redirected packets from overwhelming the supervisor module.

- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally. For more information about VLAN interfaces, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

- The VTY ACL feature restricts all traffic for all VTY lines. You cannot specify different traffic restrictions for different VTY lines. Any router ACL can be configured as a VTY ACL.

- An egress VTY ACL (an IP ACL applied to the VTY line in the outbound direction) prevents the switch from copying files using a file transfer protocol (TFTP, FTP, SCP, SFTP, etc.) unless the file transfer protocol is explicitly permitted within the egress VTY ACL.

- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.

- IP tunnels do not support ACLs or QoS policies.

- The following guidelines apply to ACLs for VXLANs:

  - Ingress port ACLs applied on a Layer 2 port for traffic in the access to a network direction (Layer 2 to Layer 3 encapsulation path) are supported on the inner payload.

  - We recommend using port ACLs on the access side to filter out traffic entering the overlay network.

  - Ingress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the network to access direction (Layer 3 to Layer 2 decapsulation path) are not supported.

  - Egress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the access to a network direction (encapsulation path) are not supported.

- Cisco Nexus 9300 and 9500 Series switches, and Cisco Nexus 9200 and 9300-EX Series switches have the following limitations for ACL options that can be used on VXLAN traffic:

  - Does not support egress port ACLs applied on a Layer 2 port for traffic in the network to access direction (decapsulation path).

  - Supports ingress VACLs applied on a VLAN for traffic in the access to a network direction (encapsulation path).

- Supports egress VACLs applied on a VLAN for traffic in the network to access direction (decapsulation path).

- Supports ingress RACLs applied on a tenant or server facing SVI for traffic in the access to network direction (encapsulation path).

- Supports egress RACLs applied on a tenant or server facing SVI for traffic in the network to access direction (decapsulation path).

- IPv6 ACL logging is not supported for egress PACL.

- IPv4 ACL logging in the egress direction is not supported.

- ACL logging for VACLs is not supported.

- ACL logging applies to port ACLs configured by the **ip port access-group** command and to router ACLs configured by the **ip access-group** command only.

- The total number of IPv4 ACL flows is limited to a user-defined maximum value to prevent DoS attacks. If this limit is reached, no new logs are created until an existing flow finishes.

- The number of syslog entries that are generated by IPv4 ACL logging is limited by the configured logging level of the ACL logging process. If the number of syslog entries exceeds this limit, the logging facility might drop some logging messages. Therefore, IPv4 ACL logging should not be used as a billing tool or as an accurate source of the number of matches to an ACL.

- Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.

- For Network Forwarding Engine (NFE)-enabled switches, ingress RACLs matching the outer header of the tunnel interface are not supported.

- If the same QoS policy and ACL are applied to multiple interfaces, the label is shared only when the QoS policy is applied with the no-stats option.

- The switch hardware does not support range checks (Layer 4 operations) in the egress TCAM. Therefore, ACL and QoS policies with a Layer 4 operations-based classification must be expanded to multiple entries in the egress TCAM.

  The switch hardware supports only up to 16 Layer 4 operands. Make sure to consider this limitation for egress TCAM space planning. For more information see the Logical Operators and Logical Operation Units, on page 295 section.

- For Cisco Nexus X96136YC-R, X9636C-RX, X9636C-RX, and X9636Q-R line cards, run the **hardware profile acl-eg-ext module all** command before applying **eg-racl-v6** configuration on a SVI or port object on an EoR switch.

- TCAM resources are shared in the following scenarios:

  - When a routed ACL is applied to multiple switched virtual interfaces (SVIs) in the ingress direction.

  - When a routed ACL is applied to multiple layer 2 interfaces in the ingress or egress direction.

- TCAM resources are not shared in the following scenarios:

  - VACL (VLAN ACL) is applied to multiple VLANs.

  - Routed ACL is applied to multiple SVIs in the egress direction.

- Access-lists based on HTTP methods are not supported on the Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9300-FXP, and 9300-GX platform switches and the 9500 switches with the X9700-EX, and X9700-FX line cards. For all these switches, you must use UDF-based ACLs.

- HTTP methods are not supported on FEX ports.

- The following guidelines and limitations apply to Cisco Nexus 9200 and 9300-EX Series switches:

  - Egress MAC ACLs are not supported.

  - Egress RACLs are not supported on an interface if the packet matches the outer header of the tunnel interface on the device where the tunnel is originating the traffic.

  - Ingress RACLs matching the outer header of the tunnel interface are not supported.

  - IP length-based matches are not supported.

  - All ACL-based features cannot be enabled at the same time.

  - 16 Layer 4 operations are supported.

  - Layer 4 operations are not supported on egress TCAM regions.

  - The MAC compression table size is 4096 + 512 overflow TCAM.

  - An overlap of MAC addresses and MAC masks is rejected.

  - The ACL log rate limiter does not have any hardware counters for transmitted or dropped packets.

  - The ACL log rate limiter is implemented at the per-TCAM entry level (instead of using aggregated rate limiting), and the default is 1 pps.

  - The Network Address Translation (NAT) exception counters are zero.

  - Only PACL redirects are supported for TAP aggregation. VACL redirects are not supported.

  - Only three of the following four features can be supported at a time: DHCPv4 snooping or relay, DHCPv6 relay, ARP snooping, VXLAN. The first three configured features take effect, but the fourth one will fail because all three bridge domain label bits are already in use.

- The following guidelines and limitations apply to Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX series switches:

  - The MAC compression table size is 4096 + 512 overflow TCAM.

  - An overlap of MAC addresses and MAC masks is rejected.

- Cisco Nexus 9504 and Cisco Nexus 9508 switches with -R line cards do not support the following TCAM:

  - All FEX related TCAM

  - All xxx-lite related TCAM region

  - Ranger related TCAM

  - All FCoE related TCAM

- TCAM carving configuration of the ing-netflow region can be performed on -FX line cards. -EX line cards have a default ing-netflow region TCAM carving of 1024 and cannot be configured otherwise. For ports on the -EX and -FX line cards, the suggested maximum for the ing-netflow region is 1024.

- On the Cisco Nexus 9200 and 9300-EX platform switches, router ACL with the ACL log option will not take into effect as the sup-redirect ACLs have higher priority for the traffic that is destined to SUP.

- On the Cisco Nexus 9300-GX platform switches, dot1q VLAN with ACL redirect supports only the VLAN IDs from 1 to 511.

  If PACL redirect or TapAgg is configured, the **switchport access vlan** *vlan-id* command supports only the vlan IDs from 1 to 511.

- For traffic destined to the FHRP VIP and ingressing on FHRP standby which matches an ACL log enabled ACE designed to permit the traffic, the Cisco Nexus 9000 Series switch drops this packet.

- For Cisco Nexus 3172TQ, 3172TQ-XL, 36180YC-R, and 3636C-R switches, when there is a SVI and subinterface matching the same VLAN tag, the traffic that gets routed out through a subinterface gets dropped if the access-list is configured on that SVI. This is due to an ASIC limitation and egress router ACL on L3 subinterfaces is not supported due to this limitation.

- Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches do not support the following on egress router ACL:

  - UDF to support ICMP Type Match.

  - ACL log-on egress

  - Egress IPv4 router ACL with additional filter option tcp/udp ports with lt/gt

  - Egress IPv4 router ACL with additional filter option tcp/udp ports with neq

  - Egress IPv4 router ACL with extra filter option tcp/udp ports with range

  - Egress IPv4 router ACL with a flag

  - Egress router ACL on an external TCAM

  - Egress PACL support

  - Statistics support

  - Label sharing

- Cisco Nexus 9500 platform switches with -R and -RX line cards have the following guidelines:

  - Atomic ACL update is supported for all the ingress ACL features except for the Multihop BFD and CoPP features.

  - Atomic ACL update is not supported for the egress ACL features.

  - Label sharing is supported only for the same policy on different interfaces within the same ASIC.

  - In Cisco NX-OS Release 9.2(3), ACL statistics are supported for the following:

    - PACL - IPv4 (including system ACL for both, internal, and external TCAM)

    - Router ACL - IPv4 (internal TCAM for both, ingress RACL-IPv4 and egress RACL-IPv4)

    - Only 2K counters are supported in the egress.

  - ACL statistics are not supported for the following:

    - BFD

- DHCP - IPv4 and IPv6

- PACL-MAC

- PACL- IPv6

- PBR - IPv4 and IPv6

- RACL-IPv6

- RACL-IPv4 when using an external TCAM

- ACL label sharing works on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 and 9300C platform switches with following limitations:

  - ACL statistics are disabled by default. However, statistics are enabled by default only for QoS policies.

  - ACL target (port / VLAN / SVI etc.,) must be on the same slice, and port.

  - Additionally, label space is shared with following features:

    - Ingress RACL, PBR and Ingress L3 QoS

    - Ingress PACL, Ingress L2 QoS

    - Egress RACL, Egress QoS

**Note** For label sharing to work, ensure that the same set of features are supported on interfaces.

- When you enable the counters for the ACL TCAM entries using the hardware profile acl-stats module *xx* command, the input discard field in the show interface is always zero. This limitation is applicable only to the Cisco Nexus 9500 platform switches with -R and -RX line cards.

- Cisco Nexus 9500 platform switches with -R and -RX line cards do not support the following:

  - Egress atomic updates

  - Egress router ACL on external TCAM

  - Egress router ACL with UDF

  - Router ACL v6 counters for both egress and ingress

  - Egress and ingress router ACL IPv6 with l4 ops

  - Egress router ACL on subinterface

  - Egress and ingress router ACL with IPv6 ICMP Type and Code

  - IPv6 ingress router ACL with tcp-flag

  - IPv4 router ACL with extra option

- In Cisco NX-OS Release 9.3(3), egress IPv4 RACLs support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:

  - TCP flags

  - ICMP Type and Code

  - ACL logs

- IPv6 Egress ACL support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:

  - Layer 4 Protocol

  - TCP flags

  - Fragment

  - ACL logs for IPv4

  - IPv6 header fields

  The following limitations are applicable for the IPv6 egress ACL:

  - Port groups and Layer 4 Operations are not supported. The ranges expand to multiple ACE entries for eg-racl-ipv6.

  - Address group defined host is not supported.

  - Counters are not supported.

  - Egress IPv6 RACL is not supported on sub-interfaces and external TCAM.

  - Atomic updates are not supported.

  - VXLAN is not supported when acl-eg-ext is enabled.

- PACL redirects are supported on Cisco Nexus 9300-GX switches. The following limitations are applicable:

  - To support PACL redirects, you must run the **mode tap-agg** command on the ingress tap interface.

  - To support the MPLS strip feature, the **mpls strip** and the **hardware acl tap-agg** commands must be configured and the switch reloaded.

  - For double tag VLAN, the range of the second VLAN is 2-510.

  - MPLS strip with dot1q VLAN is not supported.

  - The redirect port carries the tag if the incoming packet is tagged, even when the redirect port is configured as an access port.

  - TapAgg redirect is not supported for deny ACE.

- In Cisco NX-OS Release 10.1(2), PACL redirect feature is not supported in mixed mode on Cisco Nexus X9736C-FX, X9788TC-FX, and X97160YC-EX line cards.

- Egress ACL does not support traffic that is destined to the IP address of the second VLAN in inter-VLAN routing flow.

- In Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches and 93180YC-FX switches, RACLs cannot match on packets with multicast MAC destination addresses on Layer-3 interfaces. Use the **ignore routable** command when you configure the ACL to remove the routable qualifier. However, when you add ignore-routable to a RACL and apply on SVI, RACL will match with the bridged packets too.

- The Get operation provides incomplete data/no sequence number when wildcard bits are in A.B.C.D format. This is a known behavior. The Open Config model does not have srcPrefixMask/dstPrefixMask. Also, no meaningful value can be returned for prefix length because it is not possible to convert the mask to prefix length for non-contiguous mask.

- The ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).

- Beginning with Cisco NX-OS Release 9.3(9), the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platform switches.

- Beginning with Cisco NX-OS Release 10.2(3)F, the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300 Series platform switches.

- For egress RACL V6 region, you need to configure **hw profile mdb-balanced-exem**.

- From Cisco NX-OS Release 10.2(2)F, the egress PACL feature is supported on egress router ACL on Cisco Nexus 9300-GX platform switches and 93108TC-FX3P, and 93180YC-FX3 switches.

- Beginning with Cisco NX-OS Release 10.2(3)F, the egress filtering on subinterfaces feature supports Layer 3 subinterface egress router ACL on Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 platform switches.

- Beginning with Cisco NX-OS Release 10.2(3)F, the increase ACL LOU threshold feature supports configurable LOU threshold limit for ACL configuration on Cisco Nexus 9500-R platform switches.

- Beginning with Cisco NX-OS Release 10.3(1)F, ITD NAT VRF configuration is provided on Cisco Nexus 9300-GX platform switches.

- Beginning with Cisco NX-OS Release 10.3(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9808 switches.

    - Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

- Cisco Nexus 9808/9804 switches have the following limitations for ACL SUP support:

    - In ACE, match COS and match VLAN are not supported.

    - Ensure nd-na and nd-ns packets matches with IPv6 ACE.

    - TCAM carving is not supported. However, you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.

    - Central TCAM is supported. However, it is shared among both ingress and egress ACLs.

    - UDF is not supported.

- LOUs is not supported

- IPv6 fragments are not matched in egress RACL.

- L2 ACL features are not supported.

- ODM merge is not supported.

- IPv6 next header match will match the innermost next header, the pipeline is able to parse.

- Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).

- Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.

- Beginning with Cisco NX-OS Release 10.3(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9808 switches. However, UDF is not supported.

  - Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus X98900CD-A, and X9836DM-A line cards with Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards. However, UDF is not supported.

- To display ACL statistics on Cisco Nexus 9808/9804 switches, the **hardware access-list tcam per-entry-stats template racl** command has to be enabled and reload of switch is required after configuring the **hardware access-list tcam per-entry-stats template racl** command.

- Cisco Nexus 9808/9804 switches have the following limitations for CoPP support:

  - CoPP policer stats for Stage-1, Stage-2, and Stage-3 are in PPS.

**Note** CoPP Stage3 stats gets reset to zero after system switchover.

  - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.

  - Policer rates and CoS changes are supported in Custom CoPP.

  - Fabrics/FMs are not involved in in-band path.

  - CoPP Consistency checker is not supported.

  - Supported CIR minimum value is 125 PPS.

  - CIR 0 is supported.

  - There are no per entry statistics for CoPP ACL entries.

  - MACsec packets are mapped to BPDU queue.

  - Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).

- Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus 9364C-H1 switches have the following limitations for CoPP support:

    - There is only one stage of policing and CoPP policer stats for Stage-1 are in PPS

    - Policer rate and CoS changes are supported in Custom CoPP

    - Policer rates are in multiples of 572

    - CoPP Consistency checker is not supported

- Deny ACE in MAC ACL or PACL (Port ACL) with redirect option is not supported on Cisco Nexus 9000 Series switches.

- Beginning with Cisco NX-OS Release 10.3(3)F, ACL auto name completion feature is supported on Cisco Nexus 9000 Series platform switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the IP or IPv6 ACL rule priority over SUP rule on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 with 9700-EX/FX/GX line cards.

- Beginning with Cisco NX-OS Release 10.4(1)F, Security ACL is supported on the Cisco Nexus 9332D-H2R switches.

- Beginning with Cisco NX-OS Release 10.4(2)F, Security ACL is supported on the Cisco Nexus 93400LD-H1 switches.

- Beginning with Cisco Nexus Release 10.4(3)F, Security ACL is supported on the Cisco Nexus 9364C-H1 switches.

- The flexible TCAM configuration is supported on the Cisco Nexus 9332D-H2R, 9364C-H1, and 93400LD-H1 switches. Ingress and Egress regions uses 14K shared TCAM on this platform switch.

**Note**

- In Cisco Nexus 9332D-H2R, 9364C-H1, and 93400LD-H1 switches, there is no per-direction total TCAM size limit for the shared TCAM model as in Cisco Nexus 9300-FX3, GX2 switches. Limit 14K applies to the sum of both direction TCAM sizes.

- 14K is split into 28 blocks of 512 entries each. Allocation to ingress or egress direction TCAM happens in block size granularity. If the sum of ingress region sizes configured is multiple of 256, additionally 256 is allocated (and unused) for block level allocation granularity. Same applies for the sum of egress region sizes configured.

- Beginning with Cisco NX-OS Release 10.4(2)F, the new ACE configuration filter route-tag default-route is supported. This configuration enables the filtering of traffic for QoS classification when traffic matches the default route. The following guidelines and limitations are applicable for this enhancement:

    - ACL using **route-tag default-route** command is supported for class-map, QoS type only. This feature is supported for following switches:

        - Cisco Nexus 9300-FX3

        - Cisco Nexus 9300-GX

- Cisco Nexus 9300-GX2

- Cisco Nexus 93400LD-H1

- Cisco Nexus 9332D-H2R

- The **route-tag default-route** configuration is supported for IPv4 and IPv6 access-list ACEs only.

- You cannot configure both PBR ACLs and default-route on a switch. Both configurations cannot co-exist.

- If you have configured the **hardware access-list tcam pbr match-default-route** command on a switch, you cannot configure the **route-tag default-route** command in PBR policy configuration.

- Ensure that you do not have **FabricPath to VXLAN** feature configured to use the **hardware access-list tcam label ing-ifacl 6** command.

- Beginning with Cisco NX-OS Release 10.4(3)F, Class E IP address is supported for Security Group ACL with Endpoint Security Group (ESG).

- Beginning with Cisco NX-OS Release 10.5(1)F, Security Group ACL (SGACL) feature support is provided for the following features:

  - Ethernet Segment Identifier (ESI)

  - Virtual Extensible LAN Traffic Engineering (VXLAN-TE)

  - Virtual Extensible LAN Policy-Based Routing (VXLAN-PBR)

  - Cloud Security (CloudSec) for Data Center Interconnect (DCI)

  - Traffic Route Management (TRM)

# Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

**Table 18: Default IP ACL Parameters**

| Parameters | Default |
| --- | --- |
| IP ACLs | No IP ACLs exist by default |
| IP ACL entries | 1024 |
| ACL rules | Implicit rules apply to all ACLs |
| Object groups | No object groups exist by default |
| Time ranges | No time ranges exist by default |

**Related Topics**

Implicit Rules for IP and MAC ACLs, on page 293

# Configuring IP ACLs

## Creating an IP ACL

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

**Before you begin**

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>   • **ip access-list** *name*<br>   • **ipv6 access-list** *name*<br><br>**Example:**<br><br>`switch(config)# ip access-list acl-01`<br>`switch(config-acl)#` | Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters. |
| **Step 3** | (Optional) **fragments** {**permit-all** \| **deny-all**}<br><br>**Example:**<br><br>`switch(config-acl)# fragments permit-all` | Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL. |
| **Step 4** | [*sequence-number*] {**permit** \| **deny**} *protocol* {*source-ip-prefix* \| *source-ip-mask*} {*destination-ip-prefix* \| *destination-ip-mask*}<br><br>**Example:**<br><br>`switch(config-acl)# permit ip`<br>`192.168.2.0/24 any`<br><br>**Example:**<br><br>`switch(config-acl)# 10 permit ipv6 1::1`<br>`2::2 3::3 4::4` | Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295.<br><br>The **permit** and **deny** commands support many ways of identifying traffic.<br><br>For IPv4 and IPv6 access lists, you can specify a source and destination IPv4 or IPv6 prefix, which matches only on the first contiguous bits, or you can specify a source and |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | destination IPv4 or IPv6 wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch. |
| **Step 5** | (Optional) **statistics per-entry**<br><br>**Example:**<br><br>`switch(config-acl)# statistics per-entry` | Specifies that the device maintains global statistics for packets that match the rules in the ACL.<br><br>**Note** Beginning Cisco NX-OS Release 9.2(3), ACL statistics is supported on Cisco Nexus 9500 platform switches with -R line cards. This is a mandatory step if you are using the Cisco Nexus 9500 platform switches. |
| **Step 6** | **hardware profile acl-stats module** *xx*<br><br>**Example:**<br><br>`switch(config-acl)# hardware profile acl-stats module 10` | Enables counters for the ACL TCAM entries on both, the internal and external TCAM.<br><br>**Note** This command is applicable only for Cisco Nexus 9500 platform switches with -R and -RX line cards and Cisco Nexus 3636C-R and 36180YC-R switches. VLAN and SVI statistics are lost when you enable the counters. |
| **Step 7** | **reload module** *xx*<br><br>**Example:**<br><br>`switch(config)# reload module 10` | Reloads the switch.<br><br>**Note** For the Cisco Nexus 9500 platform switches, this command is optional and only those module (s) where the**hardware profile ac-stats** is applied must be reloaded. |
| **Step 8** | **ignore routeable**<br><br>**Example:**<br><br>`switch(config)# ignore routeable` | Enables the filtering of multicast traffic on Cisco Nexus 9300-EX and 9300-FX platform switches. |
| **Step 9** | (Optional) Enter one of the following commands:<br><br>• **show ip access-lists** *name*<br>• **show ipv6 access-lists** *name*<br><br>**Example:**<br><br>`switch(config-acl)# show ip access-lists acl-01` | Displays the IP ACL configuration. |
| **Step 10** | (Optional) **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

| Command or Action | Purpose |
|---|---|
| `switch(config-acl)# copy running-config`<br>`  startup-config` | |

# Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL, but 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.

### Before you begin

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 feature is especially useful for ACLs that include more than about 1000 rules.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>• **ip access-list** *name*<br>• **ipv6 access-list** *name*<br><br>**Example:**<br>`switch(config)# ip access-list acl-01`<br>`switch(config-acl)#` | Enters IP ACL configuration mode for the ACL that you specify by name. |
| **Step 3** | (Optional) [*sequence-number*] {**permit** \| **deny**} *protocol source destination*<br><br>**Example:**<br>`switch(config-acl)# 100 permit ip`<br>`192.168.2.0/24 any` | 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.<br><br>The **permit** and **deny** commands support many ways of identifying traffic. |
| **Step 4** | (Optional) [**no**] **fragments** {**permit-all** \| **deny-all**}<br><br>**Example:**<br>`switch(config-acl)# fragments permit-all` | Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL. |

| | Command or Action | Purpose |
|---|---|---|
| | | The **no** option removes fragment-handling optimization. |
| **Step 5** | (Optional) **no** {*sequence-number* \| {**permit** \| **deny**} *protocol source destination*}<br><br>**Example:**<br>`switch(config-acl)# no 80` | Removes the rule that you specified from the IP ACL.<br><br>The **permit** and **deny** commands support many ways of identifying traffic. |
| **Step 6** | (Optional) [**no**] **statistics per-entry**<br><br>**Example:**<br>`switch(config-acl)# statistics per-entry` | Specifies that the device maintains global statistics for packets that match the rules in the ACL.<br><br>The **no** option stops the device from maintaining global statistics for the ACL. |
| **Step 7** | (Optional) Enter one of the following commands:<br><br>    • **show ip access-lists** *name*<br>    • **show ipv6 access-lists** *name*<br><br>**Example:**<br>`switch(config-acl)# show ip access-lists acl-01` | Displays the IP ACL configuration. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-acl)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Changing Sequence Numbers in an IP ACL, on page 322

# Creating a VTY ACL

You can configure a VTY ACL to control access to all IPv4 or IPv6 traffic over all VTY lines in the ingress or egress direction.

### Before you begin

Set identical restrictions on all the virtual terminal lines because a user can connect to any of them.

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, which is especially useful for ACLs that include more than about 1000 rules.

**Procedure**

|        | **Command or Action** | **Purpose** |
|--------|----------------------|-------------|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | {**ip** \| **ipv6**} **access-list** *name*<br><br>**Example:**<br><br>`switch(config)# ip access-list vtyacl` | Creates an ACL and enters IP access list configuration mode for that ACL. The maximum length for the *name* argument is 64 characters. |
| **Step 3** | {**permit** \| **deny**} *protocol source destination* [**log**] [**time-range** *time*]<br><br>**Example:**<br><br>`switch(config-ip-acl)# permit tcp any any` | Creates an ACL rule that permits TCP traffic from and to the specified sources. |
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config-ip-acl)# exit`<br>`switch(config)#` | Exits IP access list configuration mode. |
| **Step 5** | **line vty**<br><br>**Example:**<br><br>`switch(config)# line vty`<br>`switch(config-line)#` | Specifies the virtual terminal and enters line configuration mode. |
| **Step 6** | {**ip** \| **ipv6**} **access-class** *name* {**in** \| **out**}<br><br>**Example:**<br><br>`switch(config-line)# ip access-class vtyacl out` | Restricts incoming or outgoing connections to and from all VTY lines using the specified ACL. The maximum length for the *name* argument is 64 characters. |
| **Step 7** | (Optional) **show** {**ip** \| **ipv6**} **access-lists**<br><br>**Example:**<br><br>`switch# show ip access-lists` | Displays the configured ACLs, including any VTY ACLs. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`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.

**Before you begin**

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 feature is especially useful for ACLs that include more than about 1000 rules.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **resequence** {**ip** \| **ipv6**} **access-list** *name starting-sequence-number increment*<br><br>**Example:**<br><br>`switch(config)# resequence access-list`<br>`ip acl-01 100 10` | Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295. |
| **Step 3** | (Optional) **show ip access-lists** *name*<br><br>**Example:**<br><br>`switch(config)# show ip access-lists`<br>`acl-01` | Displays the IP ACL configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Removing an IP ACL

You can remove an IP ACL from the device.

**Before you begin**

Ensure that you know whether the ACL is applied to an interface. The device 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 device considers the removed ACL to be empty. Use the **show ip access-lists** command or the **show ipv6 access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal```<br>```switch(config)#``` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>    • **no ip access-list** *name*<br>    • **no ipv6 access-list** *name*<br><br>**Example:**<br><br>```switch(config)# no ip access-list acl-01``` | Removes the IP ACL that you specified by name from the running configuration. |
| **Step 3** | (Optional) Enter one of the following commands:<br><br>    • **show ip access-lists** *name* **summary**<br>    • **show ipv6 access-lists** *name* **summary**<br><br>**Example:**<br><br>```switch(config)# show ip access-lists```<br>```acl-01 summary``` | Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```switch(config)# copy running-config```<br>```startup-config``` | Copies the running configuration to the startup configuration. |

# Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware. After TCAM carving, for the TCAM to qualify, you must save the configuration and reload the switch. If the switch has a faulty module, saving the configuration will take a longer time.

You can use this procedure for all Cisco Nexus 9200, 9300, and 9500 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches, except for NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module), which must use TCAM templates to configure ACL TCAM region sizes. For more information on using TCAM templates, see "Using Templates to Configure ACL TCAM Region Sizes."

✎

**Note**
- Once you apply a template (using Using Templates to Configure ACL TCAM Region Sizes, on page 334), the **hardware access-list tcam region** command in this section will not work. You must uncommit the template in order to use the command.

- The **hardware access-list tcam region** command for the Multicast PIM Bidir feature is applicable only to the Broadcom-based Cisco Nexus 9000 Series switches.

- For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **hardware access-list tcam region** *region tcam-size*<br><br>**Example:**<br><br>`switch(config)# hardware access-list tcam`<br>`region mpls 256` | Changes the ACL TCAM region size. These are the available regions:<br><br>• **n9k-arp-acl**—Configures the rate limit for arp packets entering an interface on their way to the CPU. You will have to set this rate limit per interface to ensure that arp packets conform to the configured rate.<br><br>• **arp-ether**—Configures the size of the ARP/Layer 2 Ethertype TCAM region.<br><br>• **copp**—Configures the size of the CoPP TCAM region.<br><br>• **e-flow**—Configures the size of the egress flow counters TCAM region.<br><br>• **egr-copp**—Configures the size of the egress CoPP TCAM region.<br><br>• **egr-racl**—Configures the size of the egress IPv4 or IPv6 router ACL (RACL) TCAM region.<br><br>• **egr-sup**—Configures the size of the egress supervisor TCAM region.<br><br>• **e-ipv6-qos**—Configures the size of the IPv6 egress QoS TCAM region. |

| Command or Action | Purpose |
|---|---|
| | • **e-ipv6-racl**—Configures the size of the IPv6 egress router ACL (ERACL) TCAM region. |
| | • **e-mac-qos**—Configures the size of the egress MAC QoS TCAM region. |
| | • **e-qos**—Configures the size of the IPv4 egress QoS TCAM region. |
| | • **e-qos-lite**—Configures the size of the IPv4 egress QoS lite TCAM region. |
| | • **e-racl**—Configures the size of the IPv4 egress router ACL (ERACL) TCAM region. |
| | • **fex-ifacl**—Configures the size of the FEX IPv4 port ACL TCAM region. |
| | • **fex-ipv6-ifacl**—Configures the size of the FEX IPv6 port ACL TCAM region. |
| | • **fex-ipv6-qos**—Configures the size of the FEX IPv6 port QoS TCAM region. |
| | • **fex-mac-ifacl**—Configures the size of the FEX MAC port ACL TCAM region. |
| | • **fex-mac-qos**—Configures the size of the FEX MAC port QoS TCAM region. |
| | • **fex-qos**—Configures the size of the FEX IPv4 port QoS TCAM region. |
| | • **fex-qos-lite**—Configures the size of the FEX IPv4 port QoS lite TCAM region. |
| | • **fhs**—Configures the size of the fhs TCAM region. You can configure TCAM for the fhs region on the Cisco Nexus 9300 and 9500 Series switches. |
| | • **flow**—Configures the size of the ingress flow counters TCAM region. |
| | • **ifacl**—Configures the size of the IPv4 port ACL TCAM region. |
| | • **ifacl-udf**—Configures the size of the IPv4 port ACL user-defined field (UDF) TCAM region. |
| | • **ing-ifacl**—Configures the size of the ingress IPv4, IPv6, or MAC port ACL TCAM region. |

| Command or Action | Purpose |
|---|---|
| | **Note** You can attach user-defined fields (UDFs) to the ing-ifacl TCAM region to configure UDF-based IPv4 or IPv6 port ACLs. UDF-based IPv6 port ACLs. For more information and configuration instructions, see Configuring UDF-Based Port ACLs, on page 343. |
| | • **ing-l2qos**—Configures the size of the ingress Layer 2 QoS TCAM region. |
| | • **ing-l2-span-filter**—Configures the size of the ingress Layer 2 SPAN filter TCAM region. |
| | • **ing-l3-span-filter**—Configures the size of the ingress Layer 3 and VLAN SPAN filter TCAM region. |
| | • **ing-l3-vlan-qos**—Configures the size of the ingress Layer 3, VLAN, and SVI QoS TCAM region. |
| | • **ing-netflow**—Configures the size of the NetFlow TCAM region. |
| | • **ing-racl**—Configures the size of the IPv4 or IPv6 ingress router ACL (RACL) TCAM region. |
| | • **ing-redirect**—Configures the size of the redirect TCAM region for DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client. |
| | • **ing-sup**—Configures the size of the ingress supervisor TCAM region. |
| | • **ipsg**—Configures the size of the IP source guard SMAC-IP binding TCAM region. |
| | • **ipv6-ifacl**—Configures the size of the IPv6 port ACL TCAM region. |
| | • **ipv6-l3qos**—Configures the size of the IPv6 Layer 3 QoS TCAM region. |
| | • **ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region. |
| | • **ipv6-racl**—Configures the size of the IPv6 RACL TCAM region. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **ipv6-vacl**—Configures the size of the IPv6 VACL TCAM region. |
| | | • **ipv6-vqos**—Configures the size of the IPv6 VLAN QoS TCAM region. |
| | | • **l3qos**—Configures the size of the IPv4 Layer 3 QoS TCAM region. |
| | | • **l3qos-lite**—Configures the size of the IPv4 Layer 3 QoS lite TCAM region. |
| | | • **mac-ifacl**—Configures the size of the MAC port ACL TCAM region. |
| | | • **mac-l3qos**—Configures the size of the MAC Layer 3 QoS TCAM region. |
| | | • **mac-qos**—Configures the size of the MAC port QoS TCAM region. |
| | | • **mac-vacl**—Configures the size of the MAC VACL TCAM region. |
| | | • **mac-vqos**—Configures the size of the MAC VLAN QoS TCAM region. |
| | | • **mcast_bidir**—Configures the size of the multicast PIM Bidir TCAM region. |
| | | • **mpls**—Configures the size of the static MPLS TCAM region. |
| | | • **nat**—Configures the size of the network address translation (NAT) TCAM region. |
| | | • **ns-ipv6-l3qos**—Configures the size of the IPv6 Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | | • **ns-ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | | • **ns-ipv6-vqos**—Configures the size of the IPv6 VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |

| Command or Action | Purpose |
|---|---|
| | • **ns-l3qos**—Configures the size of the IPv4 Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **ns-mac-l3qos**—Configures the size of the MAC Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **ns-mac-qos**—Configures the size of the MAC port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **ns-mac-vqos**—Configures the size of the MAC VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **ns-qos**—Configures the size of the IPv4 port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **ns-vqos**—Configures the size of the IPv4 VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). |
| | • **openflow**—Configures the size of the OpenFlow TCAM region. |
| | • **qos**—Configures the size of the IPv4 port QoS TCAM region. |
| | • **qos-lite**—Configures the size of the IPv4 port QoS lite TCAM region. |
| | • **racl**—Configures the size of the IPv4 router ACL (RACL) TCAM region. |
| | • **racl-lite**—Configures the size of the IPv4 router ACL (RACL) lite TCAM region. |
| | • **racl-udf**—Configures the size of the IPv4 router ACL (RACL) user-defined field (UDF) TCAM region. |

| Command or Action | Purpose |
|---|---|
|  | • **redirect**—Configures the size of the redirect TCAM region. |
|  | • **redirect-tunnel**—Configures the size of the redirect-tunnel TCAM region, which is used for BFD over VXLAN.<br><br>**Note**   This command is supported only if the TP_SERVICES_PKG license is installed. |
|  | • **rp-ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). |
|  | • **rp-mac-qos**—Configures the size of the MAC port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). |
|  | • **rp-qos**—Configures the size of the IPv4 port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). |
|  | • **rp-qos-lite**—Configures the size of the IPv4 port QoS lite TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). |
|  | • **sflow**—Configures the size of the sFlow TCAM region. |
|  | • **span**—Configures the size of the SPAN TCAM region. |
|  | • **svi**—Configures the size of the ingress SVI counters TCAM region. |
|  | • **vacl**—Configures the size of the IPv4 VACL TCAM region. |
|  | • **vpc-convergence**—Configures the size of the vPC convergence TCAM region. |
|  | • **vqos**—Configures the size of the IPv4 VLAN QoS TCAM region. |
|  | • **vqos-lite**—Configures the size of the IPv4 VLAN QoS lite TCAM region. |
|  | • *tcam-size*—TCAM size. The size has to a multiple of 256. If the size is more than |

| | Command or Action | Purpose |
|---|---|---|
| | | 256, it has to be multiple of 512. For FHS, the range is from 0-4096. |
| | | You can use the **no** form of this command to revert to the default TCAM region size. |
| | | **Note** You can attach IPv4 user-defined fields (UDFs) to the racl, ifacl, and vacl TCAM regions using the **hardware access-list tcam region** {**racl** \| **ifacl** \| **vacl**} **qualify udf** *udf-names* command to configure IPv4 UDF-based SPAN or ERSPAN. You can attach IPv6 UDFs to the ing-l2-span-filter and ing-l3-span-filter TCAM regions using the **hardware access-list tcam region** {**ing-ifacl** \| **ing-l2-span-filter** \| **ing-l3-span-filter**} **qualify v6udf** *v6udf-names* commands to configure IPv6 UDF-based ERSPAN. For more information and configuration instructions, see the latest *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*. |
| Step 3 | **copy running-config startup-config** <br><br>**Example:** <br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |
| Step 4 | (Optional) **show hardware access-list tcam region** <br><br>**Example:** <br>switch(config)# show hardware access-list tcam region | Displays the TCAM sizes that will be applicable on the next reload of the device. |
| Step 5 | **hardware access-list tcam label vrf-nat** <br><br>**Example:** <br>switch(config)# hardware access-list tcam label vrf-nat | Configures the ITD NAT with VRF. <br><br>**Note** Beginning with Cisco NX-OS Release 10.3(1)F, this command is supported on Cisco Nexus 9300-GX switches. |
| Step 6 | **reload** <br><br>**Example:** <br>switch(config)# reload | Reloads the device. <br><br>**Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules. |

**Example**

The following example shows how to change the size of the n9k-arp-acl TCAM region on a Cisco Nexus NFE-enabled device:

```
switch(config)#hardware access-list tcam region n9k-arp-acl 256switch(config)#copy r s
switch(config)# reload
Configuring storm-control-cpu:
switch (config)# interface ethernet 1/10switch
switch (config-if)# storm-control-cpu arp rate 150
switch (config)# show access-list storm-control-cpu arp-stats interface ethernet 1/10

slot  1
```

The following example shows how to change the size of the RACL TCAM region on a Cisco Nexus 9500 Series switch:

```
switch(config)# hardware access-list tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

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

```
switch(config)# show hardware access-list tcam region
        TCAM Region Sizes:

                            IPV4 PACL [ifacl] size =    512
                      IPV6 PACL [ipv6-ifacl] size =      0
                        MAC PACL [mac-ifacl] size =      0
                          IPV4 Port QoS [qos] size =    256
                    IPV6 Port QoS [ipv6-qos] size =      0
                      MAC Port QoS [mac-qos] size =      0
                      FEX IPV4 PACL [fex-ifacl] size =   0
              FEX IPV6 PACL [fex-ipv6-ifacl] size =      0
                  FEX MAC PACL [fex-mac-ifacl] size =    0
                  FEX IPV4 Port QoS [fex-qos] size =     0
            FEX IPV6 Port QoS [fex-ipv6-qos] size =      0
            FEX MAC Port QoS [fex-mac-qos] size =        0
                            IPV4 VACL [vacl] size =    512
                    IPV6 VACL [ipv6-vacl] size =        0
                        MAC VACL [mac-vacl] size =      0
                        IPV4 VLAN QoS [vqos] size =      0
                  IPV6 VLAN QoS [ipv6-vqos] size =       0
                    MAC VLAN QoS [mac-vqos] size =       0
                            IPV4 RACL [racl] size =    512
                    IPV6 RACL [ipv6-racl] size =        0
                  IPV4 Port QoS Lite [qos-lite] size =   0
          FEX IPV4 Port QoS Lite [fex-qos-lite] size =   0
                IPV4 VLAN QoS Lite [vqos-lite] size =     0
                IPV4 L3 QoS Lite [l3qos-lite] size =      0
                        Egress IPV4 QoS [e-qos] size =    0
                  Egress IPV6 QoS [e-ipv6-qos] size =     0
                    Egress MAC QoS [e-mac-qos] size =     0
                      Egress IPV4 VACL [vacl] size =    512
                Egress IPV6 VACL [ipv6-vacl] size =       0
                  Egress MAC VACL [mac-vacl] size =       0
                    Egress IPV4 RACL [e-racl] size =    256
                Egress IPV6 RACL [e-ipv6-racl] size =     0
```

```
                    Egress IPV4 QoS Lite [e-qos-lite] size =    0
                           IPV4 L3 QoS [l3qos] size =    0
                     IPV6 L3 QoS [ipv6-l3qos] size =    0
                       MAC L3 QoS [mac-l3qos] size =    0
                               Ingress System size =  256
                                Egress System size =  256
                                   SPAN [span] size =  256
                           Ingress COPP [copp] size =  256
              Ingress Flow Counters [flow] size =    0
               Egress Flow Counters [e-flow] size = 0
               Ingress SVI Counters [svi] size =    0
                           Redirect [redirect] size =  512
                 NS IPV4 Port QoS [ns-qos] size =  256
            NS IPV6 Port QoS [ns-ipv6-qos] size =    0
            NS MAC Port QoS [ns-mac-qos] size =    0
                 NS IPV4 VLAN QoS [ns-vqos] size =  256
            NS IPV6 VLAN QoS [ns-ipv6-vqos] size =    0
            NS MAC VLAN QoS [ns-mac-vqos] size =    0
                 NS IPV4 L3 QoS [ns-l3qos] size =  256
            NS IPV6 L3 QoS [ns-ipv6-l3qos] size =    0
            NS MAC L3 QoS [ns-mac-l3qos] size =    0
            VPC Convergence [vpc-convergence] size =  256
              IPSG SMAC-IP bind table [ipsg] size =    0
            Ingress ARP-Ether ACL [arp-ether] size =    0
       ranger+ IPV4 QoS Lite [rp-qos-lite] size =    0
                   ranger+ IPV4 QoS [rp-qos] size =  256
              ranger+ IPV6 QoS [rp-ipv6-qos] size =  256
               ranger+ MAC QoS [rp-mac-qos] size =  256
                               NAT ACL[nat] size =    0
                                   Mpls ACL size =    0
                       Ingress IPv4 N3K QoS size =    0
                       Ingress IPv6 N3K QoS size =    0
                                   MOD RSVD size =    0
                        sFlow ACL [sflow] size =    0
                          mcast bidir ACL size =    0
                                   Openflow size =    0


switch(config)# show hardware access-list tcam region
        TCAM Region Sizes:

                            IPV4 PACL [ifacl] size =    0
                      IPV6 PACL [ipv6-ifacl] size =    0
                      MAC PACL [mac-ifacl] size =    0
                          IPV4 Port QoS [qos] size =    0
                     IPV6 Port QoS [ipv6-qos] size =    0
                       MAC Port QoS [mac-qos] size =    0
                    FEX IPV4 PACL [fex-ifacl] size =    0
              FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
              FEX MAC PACL [fex-mac-ifacl] size =    0
                FEX IPV4 Port QoS [fex-qos] size =    0
           FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
           FEX MAC Port QoS [fex-mac-qos] size =    0
                            IPV4 VACL [vacl] size =    0
                      IPV6 VACL [ipv6-vacl] size =    0
                      MAC VACL [mac-vacl] size =    0
                     IPV4 VLAN QoS [vqos] size =    0
                IPV6 VLAN QoS [ipv6-vqos] size =    0
                  MAC VLAN QoS [mac-vqos] size =    0
                          IPV4 RACL [racl] size = 1536
                     IPV6 RACL [ipv6-racl] size =    0
               IPV4 Port QoS Lite [qos-lite] size =    0
       FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
               IPV4 VLAN QoS Lite [vqos-lite] size =    0
               IPV4 L3 QoS Lite [l3qos-lite] size =    0
```

```
                    Egress IPV4 QoS [e-qos] size =    0
               Egress IPV6 QoS [e-ipv6-qos] size =    0
                 Egress MAC QoS [e-mac-qos] size =    0
                   Egress IPV4 VACL [vacl] size =    0
              Egress IPV6 VACL [ipv6-vacl] size =    0
                Egress MAC VACL [mac-vacl] size =    0
                  Egress IPV4 RACL [e-racl] size =  768
            Egress IPV6 RACL [e-ipv6-racl] size =    0
       Egress IPV4 QoS Lite [e-qos-lite] size =    0
                       IPV4 L3 QoS [l3qos] size =  256
                  IPV6 L3 QoS [ipv6-l3qos] size =    0
                   MAC L3 QoS [mac-l3qos] size =    0
                           Ingress System size =  256
                            Egress System size =  256
                                SPAN [span] size =  256
                        Ingress COPP [copp] size =  256
              Ingress Flow Counters [flow] size =    0
            Egress Flow Counters [e-flow] size =    0
                Ingress SVI Counters [svi] size =    0
                      Redirect [redirect] size =  256
                 NS IPV4 Port QoS [ns-qos] size =  256
            NS IPV6 Port QoS [ns-ipv6-qos] size =    0
              NS MAC Port QoS [ns-mac-qos] size =    0
              NS IPV4 VLAN QoS [ns-vqos] size =  256
         NS IPV6 VLAN QoS [ns-ipv6-vqos] size =    0
          NS MAC VLAN QoS [ns-mac-vqos] size =    0
                NS IPV4 L3 QoS [ns-l3qos] size =  256
           NS IPV6 L3 QoS [ns-ipv6-l3qos] size =    0
             NS MAC L3 QoS [ns-mac-l3qos] size =    0
       VPC Convergence [vpc-convergence] size =  512
          IPSG SMAC-IP bind table [ipsg] size =    0
         Ingress ARP-Ether ACL [arp-ether] size =    0
```

This example shows how to revert to the default RACL TCAM region size:

```
switch(config)# no hardware profile tcam region racl 512
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

# Using Templates to Configure ACL TCAM Region Sizes

You can use create and apply custom templates to configure ACL TCAM region sizes.

For all Cisco Nexus 9200, 9300, and 9500 Series switches, you can use this procedure or the Configuring ACL TCAM Region Sizes procedure to configure ACL TCAM region sizes. However, NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module) do not support the **hardware access-list tcam region** command and must use a template to configure the ACL TCAM region size.

![Note icon]

| Note | • Once you apply a TCAM template, the **hardware access-list tcam region** command will not work. You must uncommit the template in order to use the command. |
| | • For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*. |
| | • The TCAM profile template is not supported on the C9508-FM-S fabric module. |

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **hardware profile tcam resource template** *template-name* **ref-template** {**nfe** \| **nfe2** \| {**l2-l3** \| **l3**}}<br><br>**Example:**<br>`switch(config)# hardware profile tcam`<br>`resource template SR_MPLS_CARVE`<br>`ref-template nfe2`<br>`switch(config-tcam-temp)#` | Creates a template for configuring ACL TCAM region sizes.<br><br>**nfe**—The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series.<br><br>**nfe2**—The default TCAM template for NFE2-enabled Cisco Nexus 9500 Series devices.<br><br>**l2-l3**—The default TCAM template for Layer 2 and Layer 3 configurations.<br><br>**l3**—The default TCAM template for Layer 3 configurations. |
| **Step 3** | (Optional) *region tcam-size*<br><br>**Example:**<br>`switch(config-tcam-temp)# mpls 256` | Adds any desired TCAM regions and their sizes to the template. Enter this command for each region you want to add to the template. For the list of available regions, see Configuring ACL TCAM Region Sizes. |
| **Step 4** | **exit**<br><br>**Example:**<br>`switch(config-tcam-temp)# exit`<br>`switch(config#)` | Exits the TCAM template configuration mode. |
| **Step 5** | [**no**] **hardware profile tcam resource service-template** *template-name*<br><br>**Example:**<br>`switch(config)# hardware profile tcam`<br>`resource service-template SR_MPLS_CARVE` | Applies the custom template to all line cards and fabric modules. |

| | Command or Action | Purpose |
|---|---|---|
| Step 6 | (Optional) **show hardware access-list tcam template** {**all** \| **nfe** \| **nfe2** \| **l2-l3** \| **l3** \| *template-name*}<br><br>**Example:**<br>`switch(config)# show hardware access-list tcam template SR_MPLS_CARVE` | Displays the configuration for all TCAM templates or for a specific template. |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |
| Step 8 | **reload**<br><br>**Example:**<br>`switch(config)# reload` | Reloads the device.<br><br>**Note** The configuration is effective only after you enter **copy running-config startup-config** + **reload**. |

# Configuring TCAM Carving

The default TCAM region configuration varies by platform and does not accommodate all TCAM regions. To enable any desired regions, you must decrease the TCAM size of one region and then increase the TCAM size for the desired region.

**Note** For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

**Note** Beginning with Cisco NX-OS Release 10.3(1)F, the following TCAM limitation applies on Cisco Nexus 9800 platform switches:

- TCAM carving is not supported. However you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.

- Central TCAM is supported. However it is shared among both ingress and egress ACLs.

The following tables list the default sizes for the ingress and egress TCAM regions on different platforms.

*Table 19: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9500 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1536 | 1 | 1536 |
| IPv4 Layer 3 QoS | 256 | 2 | 512 |

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| SPAN | 256 | 1 | 256 |
| CoPP | 256 | 2 | 512 |
| System | 256 | 2 | 512 |
| Redirect | 256 | 1 | 256 |
| vPC convergence | 512 | 1 | 512 |
| | | | 4K |

*Table 20: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9500 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 768 | 1 | 768 |
| System | 256 | 1 | 256 |
| | | | 1K |

*Table 21: Default TCAM Size - For Cisco Nexus 9504 and 9508 Platform switches*

| Region | Size |
|---|---|
| MAC PACL [mac-ifacl] | 1952 |
| IPV6 Port QoS [ipv6-qos] | 256 |
| PV6 L3 QoS [ipv6-l3qos] | 256 |
| SPAN [span] | 96 |
| Ingress CoPP [copp] | 128 |
| Redirect IPv4 | 2048 |
| Redirect IPv6 | 2048 |

*Table 22: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300-FX Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 2304 | 1 | 2304 |
| Layer 2 QoS | 256 | 1 | 256 |
| Layer 3/VLAN QoS | 512 | 1 | 512 |
| System | 512 | 1 | 512 |
| Layer 2 SPAN filter | 256 | 1 | 256 |
| Layer 3 SPAN filter | 256 | 1 | 256 |
| SPAN | 512 | 1 | 512 |

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| NetFlow/Analytics filter | 512 | 1 | 512 |
| | | | 5K |

*Table 23: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9300-FX Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1792 | 1 | 1792 |
| System | 256 | 1 | 256 |
| | | | 2K |

*Table 24: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300-EX Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1792 | 1 | 1792 |
| Layer 2 QoS | 256 | 1 | 256 |
| Layer 3/VLAN QoS | 512 | 1 | 512 |
| System | 512 | 1 | 512 |
| Layer 2 SPAN ACL | 256 | 1 | 256 |
| Layer 3/VLAN SPAN ACL | 256 | 1 | 256 |
| SPAN | 512 | 1 | 512 |
| | | | 4K |

*Table 25: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9300-EX Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1792 | 1 | 1792 |
| System | 256 | 1 | 256 |
| | | | 2K |

*Table 26: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 port ACL | 512 | 1 | 512 |
| IPv4 port QoS | 256 | 2 | 512 |
| IPv4 VACL | 512 | 1 | 512 |
| IPv4 RACL | 512 | 1 | 512 |
| SPAN | 256 | 1 | 256 |

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| CoPP | 256 | 2 | 512 |
| IPv4 port QoS for ACI leaf line card | 256 | 1 | 256 |
| IPv4 VLAN QoS for ACI leaf line card | 256 | 1 | 256 |
| IPv4 Layer 3 QoS for ACI leaf line card | 256 | 1 | 256 |
| System | 256 | 2 | 512 |
| Redirect | 512 | 1 | 512 |
| vPC convergence | 256 | 1 | 256 |
| | | | 4K |

*Table 27: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9300 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 VACL | 512 | 1 | 512 |
| IPv4 RACL | 256 | 1 | 256 |
| System | 256 | 1 | 256 |
| | | | 1K |

*Table 28: Default TCAM Region Configuration (Ingress) - For Layer 2-to-Layer 3 Configurations on Cisco Nexus 9200 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| Ingress NAT | 0 | 1 | 0 |
| Ingress port ACL | 256 | 1 | 256 |
| Ingress VACL | 256 | 1 | 256 |
| Ingress RACL | 1536 | 1 | 1536 |
| Ingress Layer 2 QoS | 256 | 1 | 256 |
| Ingress Layer 3 VLAN QoS | 256 | 1 | 256 |
| Ingress supervisor | 512 | 1 | 512 |
| Ingress Layer 2 ACL SPAN | 256 | 1 | 256 |
| Ingress Layer 3 ACL SPAN | 256 | 1 | 256 |
| Port-based SPAN | 512 | 1 | 512 |
| | | | 4096 |

*Table 29: Default TCAM Region Configuration (Egress) - For Layer 2-to-Layer 3 Configurations on Cisco Nexus 9200 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| Egress VACL | 256 | 1 | 256 |
| Egress RACL | 1536 | 1 | 1536 |
| Egress supervisor | 256 | 1 | 256 |
| | | | 2048 |

*Table 30: Default TCAM Region Configuration (Ingress) - For Layer 3 Configurations on Cisco Nexus 9200 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| Ingress NAT | 0 | 1 | 0 |
| Ingress port ACL | 0 | 1 | 0 |
| Ingress VACL | 0 | 1 | 0 |
| Ingress RACL | 1792 | 1 | 1792 |
| Ingress Layer 2 QoS | 256 | 1 | 256 |
| Ingress Layer 3 VLAN QoS | 512 | 1 | 512 |
| Ingress supervisor | 512 | 1 | 512 |
| Ingress Layer 2 ACL SPAN | 256 | 1 | 256 |
| Ingress Layer 3 ACL SPAN | 256 | 1 | 256 |
| Port-based SPAN | 512 | 1 | 512 |
| | | | 4096 |

*Table 31: Default TCAM Region Configuration (Egress) - For Layer 3 Configurations on Cisco Nexus 9200 Series Switches*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| Egress VACL | 0 | 1 | 0 |
| Egress RACL | 1792 | 1 | 1792 |
| Egress supervisor | 256 | 1 | 256 |
| | | | 2048 |

The following example sets the IPv6 RACL TCAM size to 256 on a Cisco Nexus 9500 Series switch. An IPv6 RACL of size 256 takes 512 entries because IPv6 is double wide.

**Note** Follow a similar procedure to modify the TCAM settings for a different region or to modify the TCAM settings on a different device.

To set the size of the ingress IPv6 RACL TCAM region on a Cisco Nexus 9500 Series switch, perform one of two options.

**Option #1**

Reduce the ingress IPv4 RACL by 1024 entries (1536 - 1024 = 512) and add an ingress IPv6 RACL with 512 entries—This option is preferred.

```
switch(config)# hardware access-list tcam region racl 512
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

*Table 32: Updated TCAM Region Configuration After Reducing the IPv4 RACL (Ingress)*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1024 | 1 | 1024 |
| IPv6 RACL | 256 | 2 | 1024[2] |
| IPv4 Layer 3 QoS | 256 | 2 | 512 |
| SPAN | 256 | 1 | 256 |
| CoPP | 256 | 2 | 512 |
| System | 256 | 2 | 512 |
| Redirect | 256 | 1 | 256 |
| vPC convergence | 512 | 1 | 512 |
| | | | 4K |

[2] 2 x 512 entry slices are allocated due to the non-availability of 256 entry slices.

**Option #2**

Remove IPv4 Layer 3 QoS by reducing its size to 0 and add an ingress IPv6 RACL—This option is available if you are not using IPv4 Layer 3 QoS.

```
switch(config)# hardware access-list tcam region l3qos 0
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

*Table 33: Updated TCAM Region Configuration After Removing Layer 3 QoS (Ingress)*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 1536 | 1 | 1536 |
| IPv6 RACL | 256 | 2 | 512 |
| IPv4 Layer 3 QoS | 0 | 2 | 0 |
| SPAN | 256 | 1 | 256 |
| CoPP | 256 | 2 | 512 |
| System | 256 | 2 | 512 |

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| Redirect | 256 | 1 | 256 |
| vPC convergence | 512 | 1 | 512 |
| | | | 4K |

To enable an egress IPv6 RACL of size 256, reduce the egress IPv4 RACL to 256 and add the egress IPv6 RACL:

```
switch(config)# hardware access-list tcam region e-racl 256
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region e-ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

*Table 34: Default TCAM Region Configuration After Reducing the IPv4 RACL (Egress)*

| Region Name | Size | Width | Total Size |
|---|---|---|---|
| IPv4 RACL | 256 | 1 | 256 |
| IPv6 RACL | 256 | 2 | 512 |
| System | 256 | 1 | 256 |
| | | | 1K |

*Table 35: Default TCAM Size - For Cisco Nexus 9800 Platform switches*

| Feature name | Size (Unidimensional) |
|---|---|
| Ingress RACLv4 | 9216 |
| Ingress QoSv4 | |
| Ingress SPAN filter v4 | |
| Egress RACLv4 | |
| Ingress SUP | |
| Ingress RACLv6 | 4608 |
| Ingress QoSv6 | |
| Ingress SPAN filter v6 | |
| Egress RACL v6 | |

**Note** Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.

After you adjust the TCAM region sizes, enter the **show hardware access-list tcam region** command to display the TCAM sizes that will be applicable on the next reload of the device.

⚠️

**Attention** To keep all modules synchronized, you must reload all line card modules or enter **copy running-config startup-config** + **reload** to reload the device. Multiple TCAM region configurations require only a single reload. You can wait until you complete all of your TCAM region configurations before you reload the device.

Depending on the configuration, you might exceed the TCAM size or run out of slices.

If you exceed the 4K ingress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM space. Please
 re-configure.
```

If you exceed the number of slices, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM slices.
Please re-configure.
```

If you exceed the 1K egress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Egress TCAM space. Please
 re-configure.
```

If TCAM for a particular feature is not configured and you try to apply a feature that requires TCAM carving, the following message appears:

```
 ERROR: Module x returned status: TCAM region is not configured. Please configure TCAM
region and retry the command.
```

✏️

**Note** The default redirect TCAM region size of 256 might not be sufficient if you are running many BFD or DHCP relay sessions. To accommodate more BFD or DHCP relay sessions, you might need to increase the TCAM size to 512 or greater.

✏️

**Note** "e-racl" tcam region size can be maximum of 16K when we have at least one "N9K-X9624D-R2" line card on a N9K-C9508 (Fretta) system.

**Related Topics**
Configuring ACL TCAM Region Sizes, on page 324

# Configuring UDF-Based Port ACLs

You can configure UDF-based port ACLs for Cisco Nexus 9200, 9300, and 9300-EX Series switches. This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to an IPv4 port ACL.

You can configure UDF-based port IPv6 ACLs for Cisco Nexus 9300-EX switches. This feature enables the device to match on the new UDFs and to apply the matching packets to an IPv6 port ACL.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **udf** *udf-name offset-base offset length*<br><br>**Example:**<br><br>`switch(config)# udf pktoff10 packet-start`<br>`10 2`<br><br>**Example:**<br><br>`switch(config)# udf pktoff10 header outer`<br>`l3 20 2` | Defines the UDF as follows:<br><br>• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.<br><br>• *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: {**packet-start** \| **header** {**outer** \| **inner** {**l3** \| **l4**}}}.<br><br>• *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.<br><br>• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.<br><br>You can define multiple UDFs, but Cisco recommends defining only required UDFs. |
| Step 3 | **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name* \| **v6udf** *v6udf-name*}<br><br>**Example:**<br><br>`switch(config)# hardware access-list tcam`<br>`region ing-ifacl qualify udf pktoff10` | Attaches the UDFs to the ing-ifacl TCAM region, which applies to IPv4 or IPv6 port ACLs.<br><br>The number of UDFs that can be attached to a TCAM region varies by platform. You can attach up to 2 UDFs for Cisco Nexus 9200 switches, up to 8 UDFs for Cisco Nexus 9300 switches, and up to 18 UDFs for IPv4 port ACLs or 7 UDFs for IPv6 port ACLs for Cisco Nexus 9300-EX switches. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes. |
| | | **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide. |
| **Step 4** | Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |
| **Step 5** | Required: **reload** **Example:** `switch(config)# reload` | Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**. |
| **Step 6** | **ip access-list** *udf-acl* **Example:** `switch(config)# ip access-list udfacl` `switch(config-acl)#` | Creates an IPv4 access control list (ACL) and enters IP access list configuration mode. |
| **Step 7** | Enter one of the following commands: <br> • **permit udf** *udf-name value mask* <br> • **permit ip** *source destination* **udf** *udf-name value mask* <br> **Example:** `switch(config-acl)# permit udf pktoff10 0x1234 0xffff` <br> **Example:** `switch(config-acl)# permit ip any any udf pktoff10 0x1234 0xffff` | Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff. <br><br> A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs. |
| **Step 8** | (Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Applying an IP ACL as a Router ACL

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

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

ACLs applied to these interface types are considered router ACLs.

> **Note** Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.

### 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 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | Enter one of the following commands:<br><br>• **interface ethernet** *slot*/*port*[**.** *number*]<br>• **interface port-channel** *channel-number*<br>• **interface vlan** *vlan-id*<br>• **interface mgmt** *port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#`<br><br>`switch(config)# interface ethernet 2/3.1`<br>`switch(config-if)#` | Enters configuration mode for the interface type that you specified. |
| Step 3 | (Optional) **encapsulation dot1q 21**<br><br>**Example:**<br><br>`switch(config-if)# encapsulation dot1q`<br>`21`<br>`switch(config-if)#` | **Note** This command is required only for Layer 3 subinterfaces. |
| Step 4 | Enter one of the following commands:<br><br>• **ip access-group** *access-list* {**in** \| **out**} | Applies an IPv4 or IPv6 ACL to the Layer 3 interface and subinterfaces for traffic flowing |

| | Command or Action | Purpose |
|---|---|---|
| | • **ipv6 traffic-filter** *access-list* {**in** | **out**}<br><br>**Example:**<br><br>`switch(config-if)# ip access-group acl1 in` | in the direction specified. You can apply one router ACL per direction. |
| Step 5 | **ip access-list match-local-traffic**<br><br>**Example:**<br><br>`switch(config-if)# ip access-list match-local-traffic` | Lists the matching traffic which is generated locally. It does not affect transit traffic through the switch. |
| Step 6 | (Optional) **show running-config aclmgr**<br><br>**Example:**<br><br>`switch(config-if)# show running-config aclmgr` | Displays the ACL configuration. |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Creating an IP ACL, on page 318

# Applying an IP ACL as a Port ACL

You can apply an IPv4 or IPv6 ACL to a Layer 2 interface, which can be a physical port or a port channel. ACLs applied to these interface types are considered port ACLs.

✏️

**Note**  If the interface is configured with the **mac packet-classify** command, you cannot apply an IP port ACL to the interface until you remove the **mac packet-classify** command from the interface configuration.

**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 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 2** | Enter one of the following commands:<br><br>• **interface ethernet** *slot*/*port*<br>• **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#` | Enters configuration mode for the interface type that you specified. |
| **Step 3** | Enter one of the following commands:<br><br>• **ip port access-group** *access-list* **in**<br>• **ipv6 port traffic-filter** *access-list* **in**<br><br>**Example:**<br>`switch(config-if)# ip port access-group`<br>`acl-l2-marketing-group in` | Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface. |
| **Step 4** | (Optional) **show running-config aclmgr**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`aclmgr` | Displays the ACL configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

**Related Topics**

# Applying an IP ACL Rule Prioritization over SUP Rule

Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for IP or IPv6 ACL which would increase the priority of ACL rule to 0 (highest) over any other SUP ACL rule that also matches on the same criteria.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>    • **ip access-list** *name*<br>    • **ipv6 access-list** *name*<br><br>**Example:**<br><br>switch(config)# ip access-list acl-01<br>switch(config-acl)# | Creates the IP or IPv6 ACL and enters the ACL configuration mode. The *name* argument can be up to 64 characters. |
| **Step 3** | [*sequence-number*] {**permit** \| **deny**} *protocol* {*source-ip-prefix* \| *source-ip-mask*} {*destination-ip-prefix* \| *destination-ip-mask*} **all**<br><br>**Example:**<br><br>For IP<br><br>switch(config-acl)# permit ip 192.168.2.0/24 any all<br><br>For IPv6<br><br>switch(config-ipv6-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4 all | Creates a rule in the IP or IPv6 ACL with an **all** keyword to prioritize the ACL rule over the SUP rule. |
| **Step 4** | Enter one of the following commands:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:**<br><br>switch(config)# interface ethernet 2/3<br>switch(config-if)# | Enters configuration mode for the interface type that you specified. |
| **Step 5** | Enter one of the following commands:<br><br>    • **ip port access-group** *access-list* **in**<br>    • **ipv6 port traffic-filter** *access-list* **in**<br><br>**Example:**<br><br>For IP<br><br>switch(config-if)# ip port access-group acl-01 in | Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface. |
| **Step 6** | (Optional) **show running-config aclmgr**<br><br>**Example:**<br><br>switch(config-if)# show running-config aclmgr | Displays the ACL configuration. |

| | Command or Action | Purpose |
|---|---|---|
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring ACL Logging

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **ip access-list** *name*<br><br>**Example:**<br>`switch(config)# ip access-list logging-test`<br>`switch(config-acl)#` | Creates an IPv4 ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters. |
| Step 3 | {**permit** \| **deny**} **ip** *source-address destination-address* **log**<br><br>**Example:**<br>`switch(config-acl)# permit ip any 10.30.30.0/24 log` | Creates an ACL rule that permits or denies IPv4 traffic matching its conditions. To enable the system to generate an informational logging message about each packet that matches the rule, you must include the **log** keyword.<br><br>The *source-address* and *destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, or **any** to designate any address. |
| Step 4 | **exit**<br><br>**Example:**<br>`switch(config-acl)# exit`<br>`switch(config)#` | Updates the configuration and exits IP ACL configuration mode. |
| Step 5 | **interface ethernet** *slot/port*<br><br>**Example:**<br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Enters interface configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 6** | **ip access-group** *name* **in**<br><br>**Example:**<br>`switch(config-if)# ip access-group logging-test in` | Enables the filtering of IPv4 traffic on an interface using the specified ACL. You can apply an ACL to inbound traffic. |
| **Step 7** | **exit**<br><br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Updates the configuration and exits interface configuration mode. |
| **Step 8** | **logging ip access-list cache interval** *interval*<br><br>**Example:**<br>`switch(config)# logging ip access-list cache interval 490` | Configures the log-update interval (in seconds) for the ACL logging process. The default value is 300 seconds. The range is from 5 to 86400 seconds. |
| **Step 9** | **logging ip access-list cache entries** *number-of-flows*<br><br>**Example:**<br>`switch(config)# logging ip access-list cache entries 8001` | Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576. |
| **Step 10** | **logging ip access-list cache threshold** *threshold*<br><br>**Example:**<br>`switch(config)# logging ip access-list cache threshold 490` | If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message. |
| **Step 11** | **logging ip access-list detailed**<br><br>**Example:**<br>`switch(config)# logging ip access-list detailed` | Enables the following information to be displayed in the output of the **show logging ip access-list cache** command: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface. |
| **Step 12** | **hardware rate-limiter access-list-log** *packets*<br><br>**Example:**<br>`switch(config)# hardware rate-limiter access-list-log 200` | Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000. |
| **Step 13** | **acllog match-log-level** *severity-level*<br><br>**Example:**<br>`switch(config)# acllog match-log-level 5` | Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging). |
| **Step 14** | (Optional) **logging ip access-list include sgt**<br><br>**Example:**<br>`switch(config)# logging ip access-list include sgt` | Enables the following SGACL information to be displayed in the output of the **show logging ip access-list cache** command: the security group tag (SGT), destination group tag (DGT), |

| | Command or Action | Purpose |
|---|---|---|
| | | source MAC (SMAC) and destination MAC (DMAC). |
| **Step 15** | (Optional) **logging ip access-list include mac**<br><br>**Example:**<br>`switch(config)# logging ip access-list`<br>`  include mac` | Enables to include MAC address as part of the ACLLOG entry to be displayed in the output of the **show logging ip access-list cache** command: source MAC (SMAC) and destination MAC (DMAC).<br><br>This command can be configured along with detailed option (**logging ip access-list detailed**), or without detailed option. |
| **Step 16** | (Optional) **show logging ip access-list cache** [**detail**]<br><br>**Example:**<br>`switch(config)# show logging ip`<br>`access-list cache` | Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces. No other information of active flows will be displayed specifically all the unsupported options.<br><br>If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.<br><br>**Note** The output format of **show logging ip access-list cache** [**detail**] will be based on the chosen optional configurations:<br><br>    • If you configure two or all optional configurations, the **include sgt** option takes the precedence over the other optional configurations.<br><br>    • For example, if the **logging ip access-list detailed** command is used along with the **logging ip access-list include sgt** command or vice-versa, the configured **include sgt** display format will be used for printing the flow information. |

# Configuring ACLs Using HTTP Methods to Redirect Requests

You can configure ACLs to intercept and redirect specific HTTP methods to a server that is connected to a specific port.

The following HTTP methods can be redirected:

- connect

- delete

- get

- head

- post

- put

- trace

### Before you begin

Enable the double-wide TCAM for the IFACL region using the **hardware access-list tcam region ifacl 512 double-wide** command. This command applies to the global configuration. Reload the switch for this configuration to take into effect.

### Procedure

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal<br>switch(config)#``` | Enters global configuration mode. |
| **Step 2** | **ip access-list** *name*<br><br>**Example:**<br><br>```switch(config)# ip access-list acl-01<br>switch(config-acl)#``` | Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters. |
| **Step 3** | [*sequence-number*] **permit** *protocol source destination* **http-method** *method* [**tcp-option-length** *length*] [**redirect** *interface*]<br><br>**Example:**<br><br>```switch(config-acl)# permit tcp 1.1.1.1/32 any http-method get``` | Configures the ACL to redirect specific HTTP methods to a server.<br><br>The following HTTP methods are supported:<br><br>• connect—Matches HTTP packets with the CONNECT method [0x434f4e4e]<br><br>• delete—Matches HTTP packets with the DELETE method [0x44454c45]<br><br>• get—Matches HTTP packets with the GET method [0x47455420] |

| | Command or Action | Purpose |
|---|---|---|
| | | • head—Matches HTTP packets with the HEAD method [0x48454144] |
| | | • post—Matches HTTP packets with the POST method [0x504f5354] |
| | | • put—Matches HTTP packets with the PUT method [0x50555420] |
| | | • trace—Matches HTTP packets with the TRACE method [0x54524143] |
| | | The **tcp-option-length** option specifies the length of the TCP options header in the packets. You can configure up to four TCP option lengths (in multiples of four bytes) in the access control entries (ACEs). The *length* range is from 0 to 40. If you do not configure this option, the length is specified as 0, and only packets without the TCP options header can match the ACE. This option allows the HTTP method to be matched even on packets that have a variable-length TCP options header. |
| | | The **redirect** option redirects an HTTP method to a server that is connected to a specific port. The HTTP redirect feature does not work on Layer 3 ports. |
| Step 4 | (Optional) **show ip access-lists** *name*<br><br>**Example:**<br>`switch(config-acl)# show ip access-lists acl-01` | Displays the IP ACL configuration. |
| Step 5 | (Optional) **show run interface** *interface slot/port*<br><br>**Example:**<br>`switch(config-acl)# show run interface ethernet 2/2` | Displays the interface configuration. |

## Example

The following example specifies a length for the TCP options header in the packets and redirects the post HTTP method to a server that is connected to port channel 4001:

```
switch(config)# ip access-list http-redirect-acl
switch(config-acl)# 10 permit tcp any any http-method get tcp-option-length 4 redirect
port-channel4001
switch(config-acl)# 20 permit tcp any any http-method post redirect port-channel4001
switch(config-acl)# statistics per-entry
switch(config)# interface Ethernet 1/33
```

```
switch(config-if)# ip port access-group http-redirect-acl in
```

# Configuring an ACL for IPv6 Extension Headers

This procedure applies only to the following devices:

- Cisco Nexus 9504 and 9508 modular chassis with these line cards: N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R

- Cisco Nexus 3600 Platform Switches (N3K-C36180YC-R and N3K-C3636C-R)

Beginning with Cisco NX-OS Release 9.3(7), if you configure an IPv6 ACL on the devices listed here, you must include a new rule for the disposition of IPv6 packets that include extension headers. For more information about IPv6 extension headers, see "Simplified IPv6 Packet Header" in NX-OS Release 9.3(x) or later of the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

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

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **ipv6 access-list** *name*<br><br>**Example:**<br>`switch(config)# ipv6 access-list acl-01`<br>`switch(config-acl)#` | Creates the IPv6 ACL and enters ACL configuration mode. |
| **Step 3** | **extension-header** {**permit-all** \| **deny-all**}<br><br>**Example:**<br>`switch(config-acl)# extension-header permit-all`<br>`switch(config-acl)#` | Choose the desired action for matched packets:<br><br>- **permit-all** — Any IPv6 packet with at least one extension header is permitted.<br><br>- **deny-all** — Any IPv6 packet with at least one extension header is dropped. |

# Verifying the IP ACL Configuration

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

| Command | Purpose |
|---|---|
| **show hardware access-list tcam region** | Displays the TCAM sizes that will be applicable on the next reload of the device. |
| **show hardware access-list tcam template** {**all** \| **nfe** \| **nfe2** \| **l2-l3** \| **l3** \| *template-name*} | Displays the configuration for all TCAM templates or for a specific template. **nfe**—The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series devices. **nfe2**—The default TCAM template for NFE2-enabled Cisco Nexus 9500 devices. **l2-l3**—The default TCAM template for Layer 2 and Layer 3 configurations. **l3**—The default TCAM template for Layer 3 configurations. |
| **show ip access-lists** | Displays the IPv4 ACL configuration. |
| **show ipv6 access-lists** | Displays the IPv6 ACL configuration. |
| **show logging ip access-list cache** [**detail**] | Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, and source interfaces. No other information of active flows will be displayed specifically all the unsupported options. If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface. |

| Command | Purpose |
|---------|---------|
| **show logging ip access-list status** | Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value. |
| **show running-config acllog** | Displays the ACL log running configuration. |
| **show running-config aclmgr** [**all**] | Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied.<br><br>**Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration. |
| **show startup-config acllog** | Displays the ACL log startup configuration. |
| **show startup-config aclmgr** [**all**] | Displays the ACL startup configuration.<br><br>**Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration. |

| Command | Purpose |
|---|---|
| **show hardware access-list interface** *ethernet X/Y* **input entries detail** | Displays the hardware ACL interface input entries' detail.<br><br>**Note**  On platforms other than 9500-R, even when the entry is expanded, the display shows the range as x y.<br><br>**Sample output for 9500-R:**<br><br>`permit tcp 100.1.1.0/24 eq 10006 100.1.1.0/24 eq 0x4e24/fffe [0]`<br><br>**Sample output for 9300-FX3S:**<br><br>`permit tcp 100.1.1.0/24 eq 10006 100.1.1.0/24 range 20004 20005 routeable 0x1 [0]` |

# Monitoring and Clearing IP ACL Statistics

To monitor or clear IP ACL statistics, use one of the commands in this table.

| Command | Purpose |
|---|---|
| **show ip access-lists** | Displays the IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, the **show ip access-lists** command output includes the number of packets that have matched each rule. |
| **show ipv6 access-lists** | Displays IPv6 ACL configuration. If the IPv6 ACL includes the **statistics per-entry** command, then the **show ipv6 access-lists** command output includes the number of packets that have matched each rule. |
| **clear ip access-list counters** | Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL. |
| **clear ipv6 access-list counters** | Clears statistics for all IPv6 ACLs or for a specific IPv6 ACL. |

# Configuration Examples for IP ACLs

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL to Ethernet interface 2/1, which is a Layer 2 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 2/1
  ip port access-group acl-01 in
```

The following example shows how to create an IPv6 ACL named acl-120 and apply it as a router ACL to Ethernet interface 2/3, which is a Layer 3 interface:

```
ipv6 access-list acl-120
  permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit tcp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
interface ethernet 2/3
  ipv6 traffic-filter acl-120 in
```

The following example shows how to create a VTY ACL named single-source and apply it on input IP traffic over the VTY line. This ACL allows all TCP traffic through and drops all other IP traffic:

```
ip access-list single-source
  permit tcp 192.168.7.5/24 any
  exit
  line vty
  ip access-class single-source in
  show ip access-lists
```

The following example shows how to configure IPv4 ACL logging:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip access-list logging-test
switch(config-acl)# permit ip any 2001:DB8:1::1/64 log
switch(config-acl)# exit
switch(config)# interface ethernet 1/1
switch(config-if)# ip access-group logging-test in
switch(config-if)# exit
switch(config)# logging ip access-list cache interval 400
switch(config)# logging ip access-list cache entries 100
switch(config)# logging ip access-list cache threshold 900
switch(config)# hardware rate-limiter access-list-log 200
switch(config)# acllog match-log-level 5
```

The following example shows how to configure a UDF-based port ACL:

```
switch# configure terminal
switch(config)# hardware access-list tcam region ing-ifacl 256
switch(config)# udf pktoff10 packet-start 10 2
switch(config)# udf pktoff20 packet-start 10 1
switch(config)# hardware access-list tcam region ing-ifacl qualify udf pktoff10 pktoff20

switch# configure terminal
switch(config)# ip access-list udfacl
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 permit ip any any udf pktoff10 0x1234 0xffff

switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ip port access-group udfacl in
switch(config-if)# switchport
switch(config-if)# no shutdown
```

The following example shows the configuration of the **route-tag default-route**:

```
switch(config)# ip access-list global-acl
switch(config-acl)# 10 permit ip any any route-tag default-route
switch(config-acl)# exit
```

```
switch(config)#class-map type qos global
switch(config-cmap-qos)#match access-group name global-acl

switch(config)#class-map type qos domestic
switch(config-cmap-qos)#match access-group name domestic-acl

switch(config)#policy-map type qos pmap
switch(config-pmap)#class global
switch(config-pmap-c)#police cir 100 mbps bc 200 ms conform transmit violate drop
switch(config-pmap)#class domestic
switch(config-pmap-c)#police cir 200 mbps bc 200 ms conform transmit violate drop

switch(config)#interface ethernet1/12
switch(config-if)#service-policy type qos input pmap

switch(config)# show running-config ipqos
!Running configuration last done at: Tue Jun 13 10:08:38 2023
!Time: Tue Jun 13 10:10:05 2023
version 10.4(2) Bios:version 01.08
class-map type qos match-all global
match access-group name global-acl
class-map type qos match-all domestic
match access-group name domestic-acl
policy-map type qos pmap
class global
police cir 100 mbps bc 200 ms conform transmit violate drop
class domestic
police cir 200 mbps bc 200 ms conform transmit violate drop
```

# About System ACLs

You can configure system ACLs on Cisco Nexus 9500 Series switches with -R and -RX line cards. With system ACLs, you can now configure a Layer 2 port ACL (PACL) on all the ports with the same access-list in the switch. Configuring system ACLs reduces the TCAM usage and also brings down the time and memory usage while the policy is being applied or modified.

See the following guidelines and limitations for configuring system ACLs:

- The system PACL is supported for Layer 2 interface only.

- Up to 10K ACEs are supported with all other basic features for the switch to come up on Cisco Nexus 9500 Series switches with -R line cards. The hardware capacity on Cisco Nexus 9500 Series switches with -RX line cards is 64K ACEs.

- You can also configure system ACLs on Cisco Nexus 3600 platform switches with N3K-C3636C-R and N3K-C36180YC-R line cards.

- Configuring IPv4 PACL TCAM region (ifacl) with anything more than the total physical TCAM capacity of -R line cards of 12k results in the power down of -R line cards only.

- ACE statistics are not yet supported for the system ACLs.

- IPv6 is not yet supported in the system ACLs.

- System ACLs are not supported on the breakout port.

- For quality of service, ACL, or TCAM carving configuration on Cisco Nexus Series switches with -R series line cards, see the Cisco Nexus 3600 NX-OS Quality of Service Configuration Guide, Release 7.x for more information.

- The non-atomic update either drops or permits all the traffic. By default, the non-atomic update drops all the traffic until the ACL update completes. The non-atomic ACL update behavior can be controlled using the **hardware access-list update default-result permit** CLI command. This CLI works only for physical ports. See the following example:

```
hardware access-list update default-result permit      => #Allows all the traffic during
 ACL updates. There may be < 10secs traffic drop.
no hardware access-list update default-result permit   => #This is the default behavior.
 It denies all the traffic during ACL updates.
```

- In Cisco NX-OS Release 9.2(2) and earlier releases, although the atomic ACL update is not supported on Cisco Nexus -R series line cards, the non-atomic update **hardware access-list update default-result** is supported on the Cisco Nexus -R series line cards.

# Carving a TCAM Region

Before configuring the system ACLs, carve the TCAM region first. Note that for configuring the ACLs less than 1k, you do not need to carve the TCAM region. See the Configuring ACL TCAM Region Sizes, on page 324 section for more information.

> ✎ **Note**    Beginning with Cisco NX-OS Release 7.0(3)F3(4) or a later release, you can configure PACL IPv4, RACL IPv4, and RACL IPv6 beyond 12k.

# Configuring System ACLs

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

**Before you begin**

Create an IPv4 ACL on the device. See Creating an IP ACL, on page 318 for more information.

**Procedure**

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

# Configuration and Show Command Examples for the System ACLs

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

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

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

Step 1: Create PACL.

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

    1000 deny any any
```

Step 2: Apply PACL into system level.

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

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

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

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

```
switch# sh ip access-lists test

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

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

switch#
```

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

```
switch# show hardware access-list tcam region
*********************************WARNING*********************************
```

```
*****************The output shows NFE tcam region info******************
***Please refer to 'show hardware access-list tcam template' for NFE2***
***********************************************************************
                        IPV4 PACL [ifacl] size = 12280
                   IPV6 PACL [ipv6-ifacl] size =     0
                    MAC PACL [mac-ifacl] size =     0
                     IPV4 Port QoS [qos] size =   640
                IPV6 Port QoS [ipv6-qos] size =   256
                 MAC Port QoS [mac-qos] size =     0
               FEX IPV4 PACL [fex-ifacl] size =     0
          FEX IPV6 PACL [fex-ipv6-ifacl] size =     0
            FEX MAC PACL [fex-mac-ifacl] size =     0
             FEX IPV4 Port QoS [fex-qos] size =     0
        FEX IPV6 Port QoS [fex-ipv6-qos] size =     0
          FEX MAC Port QoS [fex-mac-qos] size =     0
                       IPV4 VACL [vacl] size =     0
                  IPV6 VACL [ipv6-vacl] size =     0
                   MAC VACL [mac-vacl] size =     0
                   IPV4 VLAN QoS [vqos] size =     0
              IPV6 VLAN QoS [ipv6-vqos] size =     0
                MAC VLAN QoS [mac-vqos] size =     0
                       IPV4 RACL [racl] size =     0
                  IPV6 RACL [ipv6-racl] size =   128
         IPV4 Port QoS Lite [qos-lite] size =     0
    FEX IPV4 Port QoS Lite [fex-qos-lite] size =     0
       IPV4 VLAN QoS Lite [vqos-lite] size =     0
       IPV4 L3 QoS Lite [l3qos-lite] size =     0
              Egress IPV4 QoS [e-qos] size =     0
         Egress IPV6 QoS [e-ipv6-qos] size =     0
           Egress MAC QoS [e-mac-qos] size =     0
              Egress IPV4 VACL [vacl] size =     0
         Egress IPV6 VACL [ipv6-vacl] size =     0
           Egress MAC VACL [mac-vacl] size =     0
             Egress IPV4 RACL [e-racl] size =     0
        Egress IPV6 RACL [e-ipv6-racl] size =     0
       Egress IPV4 QoS Lite [e-qos-lite] size =     0
                     IPV4 L3 QoS [l3qos] size =   640
                IPV6 L3 QoS [ipv6-l3qos] size =   256
                 MAC L3 QoS [mac-l3qos] size =     0
                         Ingress System size =     0
                          Egress System size =     0
                             SPAN [span] size =    96
                   Ingress COPP [copp] size =   128
            Ingress Flow Counters [flow] size =     0
switch#
```

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

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

# Configuring 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 9000 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.

**Note**   Beginning Cisco Nexus Release 7.0(3)I5(2), the **no host IPv4-address** command is not supported. With the DME support, deletion without the no sequence command is not supported.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br><br> **Example:** <br><br> switch# configure terminal <br> switch(config)# | Enters global configuration mode. |
| **Step 2** | **object-group ip address** *name* <br><br> **Example:** <br><br> switch(config)# object-group ip address <br> ipv4-addr-group-13 <br> switch(config-ipaddr-ogroup)# | Creates the IPv4 address object group and enters IPv4 address object-group configuration mode. |
| **Step 3** | Enter one of the following commands: <br><br> • [*sequence-number*] **host** *IPv4-address* <br> • [*sequence-number*] *IPv4-address/prefix-len* <br> • [*sequence-number*] *IPv4-address network-wildcard* <br><br> **Example:** <br><br> switch(config-ipaddr-ogroup)# host <br> 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. <br><br> You can specify a prefix length for an IPv4 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address. |
| **Step 4** | Enter one of the following commands: <br><br> • **no** [*sequence-number*] <br> • **no host** *IPv4-address* <br> • **no** *IPv4-address/prefix-len* <br> • **no** *IPv4-address network-wildcard* <br><br> **Example:** | 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. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-ipaddr-ogroup)# no host 10.99.32.6` | |
| Step 5 | (Optional) **show object-group** *name* <br><br>**Example:**<br>`switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13` | Displays the object group configuration. |
| Step 6 | (Optional) **copy running-config startup-config** <br><br>**Example:**<br>`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 | **configure terminal** <br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **object-group ipv6 address** *name* <br><br>**Example:**<br>`switch(config)# object-group ipv6 address ipv6-addr-group-A7`<br>`switch(config-ipv6addr-ogroup)#` | Creates the IPv6 address object group and enters IPv6 address object-group configuration mode. |
| Step 3 | Enter one of the following commands:<br><br>• [*sequence-number*] **host** *IPv6-address*<br>• [*sequence-number*] *IPv6-address*/*prefix-len*<br>• [*sequence-number*] *IPv6-address network-wildcard*<br><br>**Example:**<br>`switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1`<br><br>**Example:**<br>`switch(config-ipv6addr-ogroup)# 10 1::1 2::2` | 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.<br><br>You can specify a prefix length for an IPv6 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 4** | Enter one of the following commands:<br><br>• **no** *sequence-number*<br>• **no host** *IPv6-address*<br>• **no** *IPv6-address*/*prefix-len*<br>• **no** *IPv6-address network-wildcard*<br><br>**Example:**<br>`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*<br><br>**Example:**<br>`switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7` | Displays the object group configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`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**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **object-group ip port** *name*<br><br>**Example:**<br>`switch(config)# object-group ip port NYC-datacenter-ports`<br>`switch(config-port-ogroup)#` | Creates the protocol port object group and enters port object-group configuration mode. |
| **Step 3** | [*sequence-number*] *operator port-number* [*port-number*]<br><br>**Example:**<br>`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:<br><br>• **eq**—Matches only the port number that you specify. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **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 numbers 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*]}<br><br>**Example:**<br>`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*<br><br>**Example:**<br>`switch(config-port-ogroup)# show`<br>`object-group NYC-datacenter-ports` | Displays the object group configuration. |
| Step 6 | (Optional) **copy running-config**<br>**startup-config**<br><br>**Example:**<br>`switch(config-port-ogroup)# copy`<br>`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**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **no object-group** {**ip address** \| **ipv6 address** \| **ip port**} *name*<br><br>**Example:**<br>`switch(config)# no object-group ip address ipv4-addr-group-A7` | Removes the specified object group. |
| **Step 3** | (Optional)  **show object-group**<br><br>**Example:**<br>`switch(config)# show object-group` | Displays all object groups. The removed object group should not appear. |
| **Step 4** | (Optional)  **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Verifying the Object-Group Configuration

To display object-group configuration information, enter one of the following commands:

| Command | Purpose |
|---|---|
| **show object-group** | Displays the object-group configuration. |
| **show** {**ip** \| **ipv6**} **access-lists** *name* [**expanded**] | Displays expanded statistics for the ACL configuration. |
| **show running-config aclmgr** | Displays the ACL configuration, including object groups. |

# Configuring Time-Ranges

## Session Manager Support for Time-Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

## Creating a Time-Range

You can create a time range on the device and add rules to it.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **time-range** *name*<br><br>**Example:**<br>`switch(config)# time-range`<br>`workday-daytime`<br>`switch(config-time-range)#` | Creates the time range and enters time-range configuration mode. |
| Step 3 | (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*<br><br>**Example:**<br>`switch(config-time-range)# periodic`<br>`monday 00:00:00 to friday 23:59:59` | Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times. |
| Step 4 | (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*<br><br>**Example:**<br>`switch(config-time-range)# periodic`<br>`weekdays 06:00:00 to 20:00:00` | Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument:<br><br>• **daily** —All days of the week.<br><br>• **weekdays** —Monday through Friday.<br><br>• **weekend** —Saturday through Sunday. |
| Step 5 | (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]<br><br>**Example:**<br>`switch(config-time-range)# absolute start`<br>`1:00 15 march 2013` | Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed. |
| Step 6 | (Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*<br><br>**Example:**<br>`switch(config-time-range)# absolute end`<br>`23:59:59 31 may 2013` | Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed. |
| Step 7 | (Optional) **show time-range** *name*<br><br>**Example:**<br>`switch(config-time-range)# show`<br>`time-range workday-daytime` | Displays the time-range configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-time-range)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Changing a Time-Range

You can add and remove rules in an existing time range. 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** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **time-range** *name*<br><br>**Example:**<br>`switch(config)# time-range workday-daytime`<br>`switch(config-time-range)#` | Enters time-range configuration mode for the specified time range. |
| **Step 3** | (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*<br><br>**Example:**<br>`switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59` | Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times. |
| **Step 4** | (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*<br><br>**Example:**<br>`switch(config-time-range)# 100 periodic weekdays 05:00:00 to 22:00:00` | Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument:<br><br>• **daily** —All days of the week.<br><br>• **weekdays** —Monday through Friday.<br><br>• **weekend** —Saturday through Sunday. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 5** | (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]<br><br>**Example:**<br>`switch(config-time-range)# absolute start 1:00 15 march 2013` | Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed. |
| **Step 6** | (Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*<br><br>**Example:**<br>`switch(config-time-range)# absolute end 23:59:59 31 may 2013` | Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed. |
| **Step 7** | (Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*}<br><br>**Example:**<br>`switch(config-time-range)# no 80` | Removes the specified rule from the time range. |
| **Step 8** | (Optional) **show time-range** *name*<br><br>**Example:**<br>`switch(config-time-range)# show time-range workday-daytime` | Displays the time-range configuration. |
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-time-range)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Removing a Time-Range

You can remove a time range from the device.

**Before you begin**

Ensure that you know whether the time range is used in any ACL rules. The device allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the ACL rule using the removed time range to be empty.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| **Step 2** | **no time-range** *name*<br><br>**Example:**<br><br>```<br>switch(config)# no time-range<br>daily-workhours<br>``` | Removes the time range that you specified by name. |
| **Step 3** | (Optional) **show time-range**<br><br>**Example:**<br><br>```<br>switch(config-time-range)# show<br>time-range<br>``` | Displays the configuration for all time ranges. The removed time range should not appear. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>```<br>switch# copy running-config<br>startup-config<br>``` | Copies the running configuration to the startup configuration. |

# Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| **Step 2** | **resequence time-range** *name starting-sequence-number increment*<br><br>**Example:**<br><br>```<br>switch(config)# resequence time-range<br>daily-workhours 100 10<br>switch(config)#<br>``` | Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. |
| **Step 3** | (Optional) **show time-range** *name*<br><br>**Example:**<br><br>```<br>switch(config)# show time-range<br>daily-workhours<br>``` | Displays the time-range configuration. |

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

# Verifying the Time-Range Configuration

To display time-range configuration information, perform one of the following tasks.

| Command | Purpose |
|---------|---------|
| **show time-range** | Displays the time-range configuration. |
| **show running-config aclmgr** | Displays ACL configuration, including all time ranges. |

# Additional References for IP ACLs

**Related Documents**

| Related Topic | Document Title |
|---------------|----------------|
| TAP aggregation | Configuring TAP Aggregation and MPLS Stripping |

# Configuring MAC ACLs

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

This chapter contains the following sections:

# About MAC ACLs

MAC ACLs are ACLs that use information in the Layer 2 header of packets to filter traffic. MAC ACLs share many fundamental concepts with IP ACLs, including support for virtualization.

# MAC Packet Classification

MAC packet classification allows you to control whether a MAC ACL that is on a Layer 2 interface applies to all traffic entering the interface, including IP traffic, or to non-IP traffic only.

| MAC Packet Classification State | Effect on Interface |
|---|---|
| Enabled | • A MAC ACL that is on the interface applies to all traffic entering the interface, including IP traffic.<br><br>• You cannot apply an IP port ACL on the interface. |
| Disabled | • A MAC ACL that is on the interface applies only to non-IP traffic entering the interface.<br><br>• You can apply an IP port ACL on the interface |

# Guidelines and Limitations for MAC ACLs

MAC ACLs have the following configuration guidelines and limitations:

- MAC ACLs apply to ingress traffic only.

- If you try to apply too many ACL entries, the configuration might be rejected.

- MAC packet classification is not supported when a MAC ACL is applied as part of a VACL.

- MAC packet classification is not supported when MAC ACLs are used as match criteria for QoS policies on Cisco Nexus 9300 Series switch 40G uplink ports.

- When you define a MAC ACL on the non EX/FX Cisco Nexus 9000 Series switches, you must define the ethertype for the traffic to be appropriately matched.

- Ethertype is required to match MAC ACL for EX/FX Cisco Nexus 9000 Series switches.

- Mac-packet classify knob is partially supported on the Cisco Nexus 9300-EX platform switches. In the absence of a direct field for marking the packet as an L2 packet, the switches match all packets with certain fields, such as src_mac, dst_mac, and vlan in the key field. However, they cannot match on the eth_type field. Therefore, if you install two rules with identical fields, except the MAC protocol number field, then the match conditions will remain identical in the hardware. Hence, although the first entry in the rule sequence will hit for all the packets for all the protocol numbers, the MAC protocol number will be a no-op when the mac-packet classify is configured.

- When you set a user-defined MAC limit using the **mac address-table limit <16-256> user-defined** command, the FHRP group limit is automatically adjusted to make the total user defined MAC limits and the FHRP limits to 490. For example, if you set the user dfefined MAC limit as 100, the FHRP limit gets reduced to 390.

- Beginning Cisco NX-OS Release 9.3(2), you can configure a user-defined MAC address limit between the range of 16–256.

- Cisco Nexus 93600CD-GX switches do not support breakout on port 1/1-24.

- A MAC access list applied to an interface will not block Bridge Protocol Data Unit (BPDU) traffic, such as Spanning Tree Protocol BPDUs.

- Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the MAC ACL rule priority over SUP rule on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 with 9700-EX/FX/GX line cards.

# Default Settings for MAC ACLs

This table lists the default settings for MAC ACL parameters.

*Table 36: Default MAC ACLs Parameters*

| Parameters | Default |
|---|---|
| MAC ACLs | No MAC ACLs exist by default |

| Parameters | Default |
|---|---|
| ACL rules | Implicit rules apply to all ACLs |

# Configuring MAC ACLs

## Creating a MAC ACL

You can create a MAC ACL and add rules to it.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **mac access-list** *name*<br><br>**Example:**<br><br>`switch(config)# mac access-list`<br>`acl-mac-01`<br>`switch(config-mac-acl)#` | Creates the MAC ACL and enters ACL configuration mode. |
| **Step 3** | {**permit** \| **deny**} *source destination-protocol*<br><br>**Example:**<br><br>`switch(config-mac-acl)# 100 permit mac`<br>`00c0.4f00.0000 0000.00ff.ffff any 0x0806` | Creates a rule in the MAC ACL.<br><br>The **permit** and **deny** commands support many ways of identifying traffic. |
| **Step 4** | (Optional) **statistics per-entry**<br><br>**Example:**<br><br>`switch(config-mac-acl)# statistics`<br>`per-entry` | Specifies that the device maintains global statistics for packets that match the rules in the ACL. |
| **Step 5** | (Optional) **show mac access-lists** *name*<br><br>**Example:**<br><br>`switch(config-mac-acl)# show mac`<br>`access-lists acl-mac-01` | Displays the MAC ACL configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-mac-acl)# copy`<br>`running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring a UDF-Based MAC ACL

You can configure UDF-based MAC access lists (ACLs) for the Cisco Nexus 9200, 9300, and 9300-EX Series switches. This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to MAC ACLs.

Beginning Cisco NX-OS Release 9.3(3), you can configure UDF-based MAC access lists (ACLs) on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **udf** *udf-name offset-base offset length*<br><br>**Example:**<br>`switch(config)# udf pktoff10`<br>`packet-start 10 2` | Defines the UDF as follows:<br><br>• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.<br><br>• *offset-base*—Specifies the UDF offset base as follows: {**packet-start**}.<br><br>• *offset*—Specifies the number of bytes offset from the offset base.<br><br>• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.<br><br>You can define multiple UDFs, but Cisco recommends defining only required UDFs. |
| **Step 3** | **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name* }<br><br>**Example:**<br>`switch(config)# hardware access-list`<br>`tcam region ing-ifacl qualify udf`<br>`pktoff10` | Attaches the UDFs to the ing-ifacl TCAM region, which applies to IPv4 or IPv6 port ACLs.<br><br>Up to 18 UDFs are supported.<br><br>**Note**    When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide. |
| **Step 4** | Required: **copy running-config startup-config** <br><br> **Example:** <br> switch(config)# copy running-config startup-config | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |
| **Step 5** | Required: **reload** <br><br> **Example:** <br> switch(config)# reload | Reloads the device. <br><br> **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**. |
| **Step 6** | **mac access-list** *udf-acl* <br><br> **Example:** <br> switch(config)# mac access-list udfacl <br> switch(config-acl)# | Creates a MAC access control list (ACL) and enters MAC ACL configuration mode. |
| **Step 7** | **permit mac** *source destination* **udf** *udf-name value mask* <br><br> **Example:** <br> switch(config-acl)# permit mac any any udf pktoff10 0x1234 0xffff | Configures the MAC ACL to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff. <br><br> A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs. |
| **Step 8** | **interface port-channel** *channel-number* <br><br> **Example:** <br> switch(config)# interface port-channel 5 <br> switch(config-if)# | Enters interface configuration mode for a Layer 2 port-channel interface. |
| **Step 9** | **mac port access-group** *udf-access-list* <br><br> **Example:** <br> switch(config-if)# mac port access-group udf-acl-01 | Applies the UDF-based MAC ACL to the interface. |
| **Step 10** | (Optional) **copy running-config startup-config** <br><br> **Example:** | Copies the running configuration to the startup configuration. |

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

# Changing a MAC ACL

You can remove a MAC ACL from the device.

### Before you begin

Use the **show mac access-lists** command with the **summary** keyword to find the interfaces on which a MAC ACL is configured.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **mac access-list** *name*<br><br>**Example:**<br>`switch(config)# mac access-list`<br>`acl-mac-01`<br>`switch(config-mac-acl)#` | Enters ACL configuration mode for the ACL that you specify by name. |
| **Step 3** | (Optional) [*sequence-number*] {**permit** \| **deny**} *source destination-protocol*<br><br>**Example:**<br>`switch(config-mac-acl)# 100 permit mac`<br>`00c0.4f00.0000 0000.00ff.ffff any 0x0806` | Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules.<br><br>The **permit** and **deny** commands support many ways of identifying traffic. |
| **Step 4** | (Optional) **no** {*sequence-number* \| {**permit** \| **deny**} *source destination-protocol*}<br><br>**Example:**<br>`switch(config-mac-acl)# no 80` | Removes the rule that you specify from the MAC ACL.<br><br>The **permit** and **deny** commands support many ways of identifying traffic. |
| **Step 5** | (Optional) [**no**] **statistics per-entry**<br><br>**Example:**<br>`switch(config-mac-acl)# statistics`<br>`per-entry` | Specifies that the device maintains global statistics for packets that match the rules in the ACL.<br><br>The **no** option stops the device from maintaining global statistics for the ACL. |
| **Step 6** | (Optional) **show mac access-lists** *name*<br><br>**Example:** | Displays the MAC ACL configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-mac-acl)# show mac access-lists acl-mac-01` | |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-mac-acl)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Changing Sequence Numbers in a MAC ACL

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **resequence mac access-list** *name starting-sequence-number increment*<br><br>**Example:**<br><br>`switch(config)# resequence mac access-list acl-mac-01 100 10` | Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify. |
| Step 3 | (Optional) **show mac access-lists** *name*<br><br>**Example:**<br><br>`switch(config)# show mac access-lists acl-mac-01` | Displays the MAC ACL configuration. |
| Step 4 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Removing a MAC ACL

You can remove a MAC ACL from the device.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **no mac access-list** *name*<br><br>**Example:**<br><br>`switch(config)# no mac access-list`<br>`acl-mac-01`<br>`switch(config)#` | Removes the MAC ACL that you specify by name from the running configuration. |
| **Step 3** | (Optional) **show mac access-lists** *name* **summary**<br><br>**Example:**<br><br>`switch(config)# show mac access-lists`<br>`acl-mac-01 summary` | Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Applying a MAC ACL as a Port ACL

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

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

**Before you begin**

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

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands: | • Enters interface configuration mode for a Layer 2 or Layer 3 interface. |

| | Command or Action | Purpose |
|---|---|---|
| | • **interface ethernet** *slot*/*port*<br>• **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#`<br><br>**Example:**<br>`switch(config)# interface port-channel`<br>`5`<br>`switch(config-if)#` | • Enters interface configuration mode for a Layer 2 or Layer 3 port-channel interface. |
| Step 3 | **mac port access-group** *access-list*<br><br>**Example:**<br>`switch(config-if)# mac port access-group`<br>`acl-01` | Applies a MAC ACL to the interface. |
| Step 4 | (Optional) **show running-config aclmgr**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`aclmgr` | Displays the ACL configuration. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Applying a MAC ACL as a VACL

You can apply a MAC ACL as a VACL.

# Enabling or Disabling MAC Packet Classification

You can enable or disable MAC packet classification on a Layer 2 interface.

### Before you begin

The interface must be configured as a Layer 2 interface.

**Note** If the interface is configured with the **ip port access-group** command or the **ipv6 port traffic-filter** command, you cannot enable MAC packet classification until you remove the **ip port access-group** and **ipv6 port traffic-filter** commands from the interface configuration.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>• **interface ethernet** *slot*/*port*<br>• **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#`<br><br>**Example:**<br>`switch(config)# interface port-channel`<br>`5`<br>`switch(config-if)#` | • Enters interface configuration mode for an Ethernet interface.<br><br>• Enters interface configuration mode for a port-channel interface. |
| **Step 3** | [**no**] **mac packet-classify**<br><br>**Example:**<br>`switch(config-if)# mac packet-classify` | Enables MAC packet classification on the interface. The **no** option disables MAC packet classification on the interface. |
| **Step 4** | (Optional) Enter one of the following commands:<br><br>• **show running-config interface ethernet** *slot*/*port*<br>• **show running-config interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`interface ethernet 2/1`<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`interface port-channel 5` | • Displays the running configuration of the Ethernet interface.<br><br>• Displays the running configuration of the port-channel interface. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Applying a MAC ACL Rule Prioritization over SUP Rule

Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for a MAC ACL which would increase the priority of ACL rule to 0 (highest) over any other SUP ACL rule that also matches on the same criteria.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | **mac access-list** *name*<br><br>**Example:**<br><br>switch(config)# mac access-list<br>acl-mac-01<br>switch(config-acl)# | Creates a MAC ACL and enters the ACL configuration mode. The *name* argument can be up to 64 characters. |
| **Step 3** | {**permit** | **deny**} *source destination-protocol* **all**<br><br>**Example:**<br><br>switch(config-mac-acl)# 100 permit<br>00c0.4f00.0000 0000.00ff.ffff any all | Creates a rule in a MAC ACL with an **all** keyword to prioritize the MAC ACL rule over the SUP rule. |
| **Step 4** | Enter one of the following commands:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:**<br><br>switch(config)# interface ethernet 2/3<br>switch(config-if)# | Enters configuration mode for the interface type that you specified. |
| **Step 5** | **mac port access-group** *access-list* **in**<br><br>**Example:**<br><br>switch(config-if)# mac port access-group<br>acl-mac-01 in | Applies a MAC ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface. |
| **Step 6** | (Optional) **show running-config aclmgr**<br><br>**Example:**<br><br>switch(config-if)# show running-config<br>aclmgr | Displays the ACL configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch(config-if)# copy running-config<br>startup-config | Copies the running configuration to the startup configuration. |

# Verifying the MAC ACL Configuration

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

| Command | Purpose |
|---------|---------|
| **show mac access-lists** | Displays the MAC ACL configuration. |
| **show running-config aclmgr** [**all**] | Displays the ACL configuration, including MAC ACLs and the interfaces to which MAC ACLs are applied.<br><br>**Note**    This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration. |
| **show startup-config aclmgr** [**all**] | Displays the ACL startup configuration.<br><br>**Note**    This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration. |

# Monitoring and Clearing MAC ACL Statistics

To monitor or clear MAC ACL statistics, use one of the commands in this table.

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

# Configuration Example for MAC ACLs

The following example shows how to create a MAC ACL named acl-mac-01 and apply it to Ethernet interface 2/1, which is a Layer 2 interface in this example:

```
mac access-list acl-mac-01
  permit 00c0.4f00.0000 0000.00ff.ffff any 0x0806
interface ethernet 2/1
  mac port access-group acl-mac-01
```

# Additional References for MAC ACLs

**Related Documents**

| Related Topic | Document Title |
|---|---|
| TAP aggregation | Configuring TAP Aggregation and MPLS Stripping |

# Configuring VLAN ACLs

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

This chapter includes the following sections:

## About VLAN ACLs

A VLAN ACL (VACL) is one application of an IP ACL or a MAC ACL. You can configure VACLs to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

## VLAN Access Maps and Entries

VACLs use access maps to contain an ordered list of one or more map entries. Each map entry associates IP or MAC ACLs to an action. Each entry has a sequence number, which allows you to control the precedence of entries.

When the device applies a VACL to a packet, it applies the action that is configured in the first access map entry that contains an ACL that permits the packet.

## 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 the normal operation of the device.

**Redirect**

Redirects the traffic to one or more specified interfaces.

**Drop**

Drops the traffic. If you specify drop as the action, you can also specify that the device logs the dropped packets.

# VACL Statistics

The 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 device does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the device maintains statistics for that VACL. This feature 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.

# Session Manager Support for VACLs

Session Manager supports the configuration of VACLs. This feature allows you to verify the ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

# Prerequisites for VACLs

VACLs have the following prerequisite:

- Ensure that the IP ACL or MAC 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 VACLs

VACLs have the following configuration guidelines:

- Cisco recommends using the Session Manager to configure ACLs. This feature allows you to verify the ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

- If you try to apply too many ACL entries, the configuration might be rejected.

- VACL redirects to SPAN destination ports are not supported.

- VACL logging is not supported.

- TCAM resources are not shared when a VACL is applied to multiple VLANs.

- Cisco Nexus 9200 and 9300-EX Series switches support the VACL redirect option. The redirect is permitted to one physical or port-channel interface.

- VACLs are not supported on Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636C-RX, and N9K-X9636Q-R line cards.

- Deny statements are not supported on VACLs. Alternatively, you can use permit statements with the action 'drop' to achieve a similar outcome.

- When configuring a VACL with the "redirect" option, the interface that you define as the redirect interface, must be configured as a member of the VLAN which you apply this VACL to. This VLAN must also be in the forwarding state on this interface for the redirection to work. If these conditions are not met, then the switch will drop the packets which are matched by the VACL.

- To clear VACL counters, you must ensure that you have active VLAN filters configured.

- Beginning with Cisco NX-OS Release 10.1(2), VACL is supported on the N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.

- Beginning with Cisco NX-OS Release 10.2(1q)F, VACL, DACL, and CoPP are supported on the N9K-C9332D-GX2B platform switches.

The following guidelines apply to VACLs for VXLANs:

- VACLs applied on a VXLAN VLAN in the access to network direction (Layer 2 to Layer 3 encapsulation path) are supported on the inner payload.

- We recommend using VACLs on the access side to filter out traffic entering the overlay network.

- Egress VACLs for decapsulated VXLAN traffic are not supported.

# Default Settings for VACLs

This table lists the default settings for VACL parameters.

**Table 37: Default VACL Parameters**

| Parameters | Default |
| --- | --- |
| VACLs | No IP ACLs exist by default |
| ACL rules | Implicit rules apply to all ACLs |

# Configuring VACLs

## Creating a VACL or Adding a VACL Entry

You can create a VACL or add entries to an existing VACL. In both cases, you create a VACL entry, which is a VLAN access-map entry that associates one or more ACLs with an action to be applied to the matching traffic.

### Before you begin

Ensure that the ACLs that you want to use in the VACL exist and are configured to filter traffic in the manner that you need for this application.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **vlan access-map** *map-name* [*sequence-number*]<br><br>**Example:**<br><br>`switch(config)# vlan access-map`<br>`acl-mac-map`<br>`switch(config-access-map)#` | Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it.<br><br>If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map. |
| **Step 3** | Enter one of the following commands:<br><br>    • **match** {**ip** \| **ipv6**} **address** *ip-access-list*<br>    • **match mac address** *mac-access-list*<br><br>**Example:**<br><br>`switch(config-access-map)# match mac`<br>`address acl-ip-lab`<br><br>**Example:**<br><br>`switch(config-access-map)# match mac`<br>`address acl-mac-01` | Specifies an ACL for the access-map entry. |
| **Step 4** | **action** {**drop** \| **forward** \| **redirect**}<br><br>**Example:**<br><br>`switch(config-access-map)# action forward`<br><br>**Example:** | Specifies the action that the device applies to traffic that matches the ACL.<br><br>The **action** command supports the **drop**, **forward**, and **redirect** options. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-access-map)# vlan access-map vacl1`<br>`switch(config-access-map)# action redirect e1/1`<br>`switch(config-access-map)# action redirect po100` | |
| **Step 5** | (Optional) [**no**] **statistics per-entry**<br>**Example:**<br>`switch(config-access-map)# statistics per-entry` | Specifies that the device maintains global statistics for packets that match the rules in the VACL.<br>The **no** option stops the device from maintaining global statistics for the VACL. |
| **Step 6** | (Optional) **show running-config aclmgr**<br>**Example:**<br>`switch(config-access-map)# show running-config aclmgr` | Displays the ACL configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-access-map)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Removing a VACL or a VACL Entry

You can remove a VACL, which means that you will delete the VLAN access map.

You can also remove a single VLAN access-map entry from a VACL.

### Before you begin

Ensure that you know whether the VACL is applied to a VLAN. The device allows you to remove VACLs that are currently applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the device considers the removed VACL to be empty.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **no vlan access-map** *map-name* [*sequence-number*]<br>**Example:** | Removes the VLAN access map configuration for the specified access map. If you specify the *sequence-number* argument and the VACL contains more than one entry, the command removes only the entry specified. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# no vlan access-map acl-mac-map 10 | |
| **Step 3** | (Optional) **show running-config aclmgr**<br><br>**Example:**<br>switch(config)# show running-config aclmgr | Displays the ACL configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

### Before you begin

If you are applying a VACL, ensure that the VACL exists and is configured to filter traffic in the manner that you need for this application.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | [**no**] **vlan filter** *map-name* **vlan-list** *list*<br><br>**Example:**<br>switch(config)# vlan filter acl-mac-map vlan-list 1-20,26-30<br>switch(config)# | Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. |
| **Step 3** | (Optional) **show running-config aclmgr**<br><br>**Example:**<br>switch(config)# show running-config aclmgr | Displays the ACL configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Verifying the VACL Configuration

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

| Command | Purpose |
|---|---|
| **show running-config aclmgr** [**all**] | Displays the ACL configuration, including the VACL-related configuration.<br><br>**Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration. |
| **show startup-config aclmgr** [**all**] | Displays the ACL startup configuration.<br><br>**Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration. |
| **show vlan filter** | Displays information about VACLs that are applied to a VLAN. |
| **show vlan access-map** | Displays information about VLAN access maps. |

# Monitoring and Clearing VACL Statistics

To monitor or clear VACL statistics, use one of the commands in this table.

| Command | Purpose |
|---|---|
| **show vlan access-list** | Displays the VACL configuration. If the VLAN access-map includes the **statistics per-entry** command, the **show vlan access-list** command output includes the number of packets that have matched each rule. |
| **clear vlan access-list counters** | Clears statistics for VACLs. |

# Configuration Example for VACLs

The following example shows how to configure a VACL to forward traffic permitted by a MAC ACL named acl-mac-01 and how to apply the VACL to VLANs 50 through 82:

```
conf t
vlan access-map acl-mac-map
  match mac address acl-mac-01
  action forward
vlan filter acl-mac-map vlan-list 50-82
```

# Additional References for VACLs

**Related Documents**

| Related Topic | Document Title |
| --- | --- |
| QoS configuration | *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide* |

**CHAPTER 15**

# Configuring Port Security

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

This chapter includes the following sections:

## About Port Security

Port security allows you to configure Layer 2 physical interfaces and Layer 2 port-channel interfaces to allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

**Note**  Unless otherwise specified, the term *interface* refers to both physical interfaces and port-channel interfaces; likewise, the term *Layer 2 interface* refers to both Layer 2 physical interfaces and Layer 2 port-channel interfaces.

## Secure MAC Address Learning

The process of securing a MAC address is called learning. A MAC address can be a secure MAC address on one interface only. For each interface on which you enable port security, the device can learn a limited number

of MAC addresses by the static or dynamic methods. The way that the device stores secure MAC addresses varies depending upon how the device learned the secure MAC address.

# Static Method

The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are unaffected if the device restarts.

A static secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address from the configuration.

- You configure the interface to act as a Layer 3 interface.

Adding secure addresses by the static method is not affected by whether dynamic address learning is enabled.

# Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- The device restarts

- The interface restarts

- The address reaches the age limit that you configured for the interface

- You explicitly remove the address

- You configure the interface to act as a Layer 3 interface

# Sticky Method

If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning, but the device stores addresses learned by this method in nonvolatile RAM (NVRAM). As a result, addresses learned by the sticky method persist through a device restart. Sticky secure MAC addresses do not appear in the running configuration of an interface.

Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, the device stops dynamic learning and performs sticky learning instead. If you disable sticky learning, the device resumes dynamic learning.

A sticky secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address

- You configure the interface to act as a Layer 3 interface

# Dynamic Address Aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 0 to 1440 minutes, where 0 disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The 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.

**Note** This feature is supported only on Cisco Nexus 9200 and 9300-EX Series switches.

**Absolute**

The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

**Note** When the absolute aging time is configured, MAC aging occurs even when the traffic from the source MAC is flowing. However, during MAC aging and re-learn, there could be a transient traffic drop.

# Secure MAC Address Maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: static or dynamic.

**Tip** To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

The following three limits can determine how many secure MAC addresses are permitted on an interface:

**Device Maximum**

The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

**Interface Maximum**

You can configure a maximum number of 1025 secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Interface maximums cannot exceed the device maximum.

**VLAN Maximum**

You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. A VLAN maximum cannot exceed the configured interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first.

# Security Violations and Actions

Port security triggers security violations when either of the following events occurs:

**MAC Count Violation**

Ingress traffic arrives at an interface from a nonsecure MAC address, and learning the address would exceed the applicable maximum number of secure MAC addresses.

When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

- VLAN 1 has a maximum of five addresses
- The interface has a maximum of ten addresses

The device detects a violation when any of the following occurs:

- The device has learned five addresses for VLAN 1, and inbound traffic from a sixth address arrives at the interface in VLAN 1.
- The device has learned ten addresses on the interface, and inbound traffic from an eleventh address arrives at the interface.

The possible actions that the device can take are as follows:

**Shutdown**

Shuts down the interface that received the packet triggering the violation. The interface is error disabled. This action is the default. After you reenable the interface, it retains its port security configuration, including its secure MAC addresses.

You can use the **errdisable** global configuration command to configure the device to reenable the interface automatically if a shutdown occurs, or you can manually reenable the interface by entering the **shutdown** and **no shutdown** interface configuration commands.

**Restrict**

Drops ingress traffic from any nonsecure MAC addresses.

The device keeps a count of the number of dropped MAC addresses, which is called the security violation count. Address learning continues until the maximum security violations have occurred on the interface. Traffic from addresses learned after the first security violation is dropped.

**MAC Move Violation**

Ingress traffic from a secure MAC address arrives at a different interface in the same VLAN as the interface on which the address is secured.

You see a mac move notification only when the the logging level of Layer2 Forwarding Module (L2FM) is increased to 4 or 5

When a MAC move violation occurs, the device increments the security violation counter for the interface, and irrespective of the violation mode configured, the interface is error disabled. If the violation mode is configured as Restrict or Protect, the violation is logged in the system log.

Because a MAC move violation results in the interface being error disabled, irrespective of the violation mode configured, we recommend using the **errdisable** command to enable automatic errdisable recovery.

# Port Security and Port Types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

### Access Ports

You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN. VLAN maximums are not useful for access ports.

### Trunk Ports

You can configure port security on interfaces that you have configured as Layer 2 trunk ports. The device allows VLAN maximums only for VLANs associated with the trunk port.

### SPAN Ports

You can configure port security on SPAN source ports but not on SPAN destination ports.

### Ethernet Port Channels

You can configure port security on Layer 2 Ethernet port channels in either access mode or trunk mode.

**Note** Port security is supported for FEX interfaces only in non-vPC deployments on Cisco Nexus 9300-EX/FX/FX2/FX3 Series switches. Beginning with Cisco NX-OS Release 9.3(5), Nexus 9300-FX3 Series switches are supported.

# Port Security and Port-Channel Interfaces

Port security is supported on Layer 2 port-channel interfaces. Port security operates on port-channel interfaces in the same manner as on physical interfaces, except as described in this section.

### General Guidelines

Port security on a port-channel interface operates in either access mode or trunk mode. In trunk mode, the MAC address restrictions enforced by port security apply to all member ports on a per-VLAN basis.

Enabling port security on a port-channel interface does not affect port-channel load balancing.

Port security does not apply to port-channel control traffic passing through the port-channel interface. Port security allows port-channel control packets to pass without causing security violations. Port-channel control traffic includes the following protocols:

- Port Aggregation Protocol (PAgP)

- Link Aggregation Control Protocol (LACP)

- Inter-Switch Link (ISL)

- IEEE 802.1Q

### Configuring Secure Member Ports

The port security configuration of a port-channel interface has no effect on the port security configuration of member ports.

### Adding a Member Port

If you add a secure interface as a member port of a port-channel interface, the device discards all dynamic secure addresses learned on the member port but retains all other port-security configuration of the member port in the running configuration. Static secure MAC addresses learned on the secure member port are also stored in the running configuration rather than NVRAM.

If port security is enabled on the member port and not enabled on the port-channel interface, the device warns you when you attempt to add the member port to the port-channel interface. You can use the **force** keyword with the **channel-group** command to forcibly add a secure member port to a nonsecure port-channel interface.

While a port is a member of a port-channel interface, you cannot configure port security on the member port. To do so, you must first remove the member port from the port-channel interface.

### Removing a Member Port

If you remove a member port from a port-channel interface, the device restores the port security configuration of the member port. Static secure MAC addresses that were learned on the port before you added it to the port-channel interface are restored to NVRAM and removed from the running configuration.

**Note**     To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

### Removing a Port-Channel Interface

If you remove a secure port-channel interface, the following occurs:

- The device discards all secure MAC addresses learned for the port-channel interface, including static secure MAC addresses learned on the port-channel interface.

- The device restores the port-security configuration of each member port. The static secure MAC addresses that were learned on member ports before you added them to the port-channel interface are restored to NVRAM and removed from the running configuration. If a member port did not have port security enabled prior to joining the port-channel interface, port security is not enabled on the member port after the port-channel interface is removed.

**Note**     To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

**Disabling Port Security**

If port security is enabled on any member port, the device does not allow you to disable port security on the port-channel interface. To do so, remove all secure member ports from the port-channel interface first. After disabling port security on a member port, you can add it to the port-channel interface again, as needed.

# Port Type Changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

**Access Port to Trunk Port**

When you change a Layer 2 interface from an access port to a trunk port, the device drops all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN.

**Switched Port to Routed Port**

When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

**Routed Port to Switched Port**

When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

# Prerequisites for Port Security

Port security has the following prerequisites:

- You must globally enable port security for the device that you want to protect with port security.

# Default Settings for Port Security

This table lists the default settings for port security parameters.

| Parameters | Default |
|---|---|
| Port security enablement globally | Disabled |
| Port security enablement per interface | Disabled |
| MAC address learning method | Dynamic |
| Interface maximum number of secure MAC addresses | 1 |
| Security violation action | Shutdown |

# Guidelines and Limitations for Port Security

When configuring port security, follow these guidelines:

- Port security does not support switched port analyzer (SPAN) destination ports.

- Port security does not depend upon other features.

- Port security is supported for FEX interfaces only in non-vPC deployments on Cisco Nexus 9300-EX Series switches.

- Beginning with Cisco NX-OS Release 10.2(1)F, disabling the USB Port is supported on Cisco NX-OS switches. To disable or enable the USB ports, use the **[no] port usb disable** command.

- After configuring the association between the primary and secondary VLANs and deleting the association, all static MAC addresses that were created on the primary VLANs remain on the primary VLAN only.

> **Note**    In some cases, the configuration is accepted with no error messages, but the commands have no effect.

After configuring the association between the primary and secondary VLANs:

- Static MAC addresses for the secondary VLANs cannot be created.

- Dynamic MAC addresses that learned the secondary VLANs are aged out.

# Guidelines and Limitations for Port Security on vPCs

Apart from the guidelines and limitations for port security, check that you can meet the following guidelines and limitations for port security on vPCs:

- Port security is not supported on FEX interfaces in vPC deployments.

- You must enable port security globally on both vPC peers in a vPC domain.

- You must enable port security on the vPC interfaces of both vPC peers.

- You must configure a static secure MAC address on the primary vPC peer. The static MAC address is synchronized with the secondary vPC peer. You can also configure a static secure MAC address on the secondary peer. The second static MAC address appears in the secondary vPC configuration but does not take affect.

- You must ensure that the maximum MAC count value remains the same for both primary and secondary vPC ports.

- On a secondary vPC port, there is no limit check for static MACs configured. Cisco recommends that you configure the same number of static MACs on a secondary vPC port as defined in the maximum MAC count.

- All learned MAC addresses are synchronized between vPC peers.

- Both vPC peers can be configured using the dynamic or static MAC address learning method. Cisco recommends that you configure both vPC peers using the same method. This helps prevent port shut down (errDisabled state) in certain cases, such as a vPC role change.

- Dynamic MAC addresses are dropped only after the age limit is reached on both vPC peers.

- You set the maximum number of secure MAC addresses on the primary vPC switch. The primary vPC switch does the count validation and disregards any maximum number settings on the secondary switch.

- You must configure the violation action on the primary vPC. When a security violation is triggered, the security action defined on the primary vPC switch occurs.

- You can use the **show vpc consistency-parameters id** command to verify that the configuration is correct on both vPC peers.

- While a switch undergoes an in-service software upgrade (ISSU), port security operations are stopped on its peer switch. The peer switch does not learn any new MAC addresses, and MAC moves occurring during this operation are ignored. When the ISSU is complete, the peer switch is notified and normal port security functionality resumes.

- ISSU to higher versions is supported; however, ISSU to lower versions is not supported.

# Configuring Port Security

## Enabling or Disabling Port Security Globally

You can enable or disable port security globally on a device. By default, port security is disabled globally.

When you disable port security, all port security configuration on the interface is ineffective. When you disable port security globally, all port security configuration is lost.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | [**no**] **feature port-security**<br><br>**Example:**<br><br>`switch(config)# feature port-security` | Enables port security globally. The **no** option disables port security globally. |
| Step 3 | (Optional) **show port-security**<br><br>**Example:**<br><br>`switch(config)# show port-security` | Displays the status of port security. |

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

# Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

When you disable port security on an interface, all switchport port security configuration for the interface is lost.

### Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`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**<br><br>**Example:**<br>`switch(config-if)# switchport` | Configures the interface as a Layer 2 interface. |
| **Step 4** | [**no**] **switchport port-security**<br><br>**Example:** | Enables port security on the interface. The **no** option disables port security on the interface. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config-if)# switchport<br>port-security | |
| **Step 5** | (Optional) **show running-config port-security**<br><br>**Example:**<br>switch(config-if)# show running-config<br>port-security | Displays the port security configuration. |
| **Step 6** | (Optional) **copy running-config<br>startup-config**<br><br>**Example:**<br>switch(config-if)# copy running-config<br>startup-config | Copies the running configuration to the startup<br>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**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>  • **interface ethernet** *slot*/*port*<br>  • **interface port-channel** *channel-number*<br><br>**Example:**<br>switch(config)# interface ethernet 2/1<br>switch(config-if)# | Enters interface configuration mode for the<br>interface that you want to configure with sticky<br>MAC address learning. |
| **Step 3** | **switchport**<br><br>**Example:**<br>switch(config-if)# switchport | Configures the interface as a Layer 2 interface. |
| **Step 4** | [**no**] **switchport port-security mac-address<br>sticky**<br><br>**Example:** | Enables sticky MAC address learning on the<br>interface. The **no** option disables sticky MAC<br>address learning. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config-if)# switchport port-security mac-address sticky | |
| Step 5 | (Optional) **show running-config port-security**<br><br>**Example:**<br><br>switch(config-if)# show running-config port-security | Displays the port security configuration. |
| Step 6 | (Optional)  **copy running-config startup-config**<br><br>**Example:**<br><br>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**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 2 | Enter one of the following commands:<br><br>• **interface ethernet** *slot*/*port*<br>• **interface port-channel** *channel-number*<br><br>**Example:**<br><br>switch(config)# interface ethernet 2/1<br>switch(config-if)# | Enters interface configuration mode for the interface that you specify. |

| | Command or Action | Purpose |
|---|---|---|
| Step 3 | [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]<br><br>**Example:**<br>`switch(config-if)# switchport`<br>`port-security mac-address 0019.D2D0.00AE` | Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on. |
| Step 4 | (Optional) **show running-config port-security**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`port-security` | Displays the port security configuration. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`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**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | Enter one of the following commands:<br>　• **interface ethernet** *slot*/*port*<br>　• **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`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*<br><br>**Example:**<br>`switch(config-if)# no switchport`<br>`port-security mac-address 0019.D2D0.00AE` | Removes the static secure MAC address from port security on the current interface. |
| Step 4 | (Optional) **show running-config port-security**<br><br>**Example:** | Displays the port security configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-if)# show running-config port-security` | |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Removing a Sticky Secure MAC Address

You can remove a sticky secure MAC address, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

### Before you begin

You must have enabled port security globally.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>  • **interface ethernet** *slot*/*port*<br>  • **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`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**<br><br>**Example:**<br>`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*<br><br>**Example:**<br>`switch(config-if)# clear port-security dynamic address 0019.D2D0.02GD` | Removes the dynamic secure MAC address that you specify. |
| **Step 5** | (Optional) **show port-security address interface** {**ethernet** *slot*/*port* \| **port-channel** *channel-number*} | Displays secure MAC addresses. The address that you removed should not appear. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br><br>`switch(config)# show port-security`<br>`address interface ethernet 2/1` | |
| **Step 6** | (Optional) **switchport port-security mac-address sticky**<br><br>**Example:**<br><br>`switch(config-if)# switchport`<br>`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**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **clear port-security dynamic** {**interface ethernet** *slot*/*port* | **address** *address*} [**vlan** *vlan-ID*]<br><br>**Example:**<br><br>`switch(config)# clear port-security`<br>`dynamic interface ethernet 2/1` | Removes dynamically learned, secure MAC addresses, as specified.<br><br>If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify.<br><br>If you use the **address** keyword, you remove the single, dynamically learned address that you specify.<br><br>Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN. |
| **Step 3** | (Optional) **show port-security address**<br><br>**Example:**<br><br>`switch(config)# show port-security`<br>`address` | Displays secure MAC addresses. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

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

# Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of addresses is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

**Note** When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the device rejects the command. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

### Before you begin

You must have enabled port security globally.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>　• **interface ethernet** *slot*/*port*<br>　• **interface port-channel** *channel-number*<br><br>**Example:**<br><br>switch(config)# interface ethernet 2/1<br>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*]<br><br>**Example:**<br><br>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.<br><br>If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **show running-config port-security**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`port-security` | Displays the port security configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`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**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`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**}<br><br>**Example:**<br>`switch(config-if)# switchport`<br>`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. |
| **Step 4** | [**no**] **switchport port-security aging time** *minutes* | Configures the number of minutes that a dynamically learned MAC address must age |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br><br>switch(config-if)# switchport port-security aging time 120 | 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).<br><br>**Note**    For Cisco Nexus 9200 and 9300-EX Series switches, up to 2 minutes might be added to the configured aging time. For example, if you set the aging time to 10 minutes, the age out occurs between 10 and 12 minutes after traffic stops. |
| **Step 5** | (Optional) **show running-config port-security**<br><br>**Example:**<br><br>switch(config-if)# show running-config port-security | Displays the port security configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>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**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | Enter one of the following commands:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:** | Enters interface configuration mode for the interface that you want to configure with a security violation action. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
|  | `switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` |  |
| **Step 3** | [**no**] **switchport port-security violation** {**protect** \| **restrict** \| **shutdown**}<br><br>**Example:**<br>`switch(config-if)# switchport`<br>`port-security violation restrict` | Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface. |
| **Step 4** | (Optional) **show running-config port-security**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`port-security` | Displays the port security configuration. |
| **Step 5** | (Optional)  **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Verifying the Port Security Configuration

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

| **Command** | **Purpose** |
|---|---|
| **show running-config port-security** | Displays the port security configuration. |
| **show port-security** | Displays the port security status of the device. |
| **show port-security interface** | Displays the port security status of a specific interface. |
| **show port-security address** | Displays secure MAC addresses. |
| **show vpc consistency-parameters vpc id** | Verifies configuration on both vPC peers. |

# Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses.

# Configuration Example for Port Security

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

# Configuration Examples for Port Security in a vPC Domain

The following example shows how to enable and configure port security on vPC peers in a vPC domain. The first switch is the primary vPC peer and the second switch is the secondary vPC peer. Before configuring port security on the switches, create the vPC domain and check that the vPC peer-link adjacency is established.

## Example: Configuring Port Security on an Orphan Port

```
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)# feature port-security
secondary_switch(config)# int e3/1
secondary_switch(config-if)# switchport port-security
secondary_switch(config-if)# switchport port-security max 1025
secondary_switch(config-if)# switchport port-security violation restrict
secondary_switch(config-if)# switchport port-security aging time 4
secondary_switch(config-if)# switchport port-security aging type absolute
secondaryy_switch(config-if)# switchport port-security mac sticky
secondary_switch(config-if)# switchport port-security mac-address 0.0.1 vlan 101
secondary_switch(config-if)# switchport port-security mac-address 0.0.2 vlan 101
secondary_switch(config-if)# copy running-config startup-config
```

## Example: Configuring Port Security on the vPC Leg

```
primary_switch(config)# feature port-security
primary_switch(config-if)# int po10
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)# vpc 10
primary_switch(config-if)# copy running-config startup-config
```

```
secondary_switch(config)# feature port-security
secondary_switch(config)# int po10
secondary_switch(config-if)# switchport port-security
secondary_switch(config-if)# switchport port-security max 1025
secondary_switch(config-if)# switchport port-security violation restrict
secondary_switch(config-if)# switchport port-security aging time 4
secondary_switch(config-if)# switchport port-security aging type absolute
secondaryy_switch(config-if)# switchport port-security mac sticky
secondary_switch(config-if)# switchport port-security mac-address 0.0.1 vlan 101
secondary_switch(config-if)# switchport port-security mac-address 0.0.2 vlan 101
secondary_switch(config-if)# vpc 10
secondary_switch(config-if)# copy running-config startup-config
```

# Additional References for Port Security

### Related Documents

| Related Topic | Document Title |
|---------------|----------------|
| Layer 2 switching | *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide* |

### MIBs

Cisco NX-OS provides read-only SNMP support for port security.

| MIBs | MIBs Link |
|------|-----------|
| CISCO-PORT-SECURITY-MIB<br><br>**Note** Traps are supported for notification of secure MAC address violations. | To locate and download MIBs, go to the following URL:<br><br>https://snmp.cloudapps.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2 |

# Port Security Support for VXLAN EVPN

This section describes how to configure Port Security for VXLAN EVPN.

# Guidelines and Limitations for Port Security Support for VXLAN EVPN

The following are the guidelines and limitations for Port Security support for VXLAN EVPN:

- Beginning with Cisco NX-OS Release 10.3(3)F, the L2 port security feature is supported on VXLAN BGP EVPN (single VTEP) for Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, 9408, 9332C, 9364C switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards with the following limitations:

  - Only a Single VTEP solution is supported. However, secure MAC mobility is not supported on a VXLAN environment.

- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 underlay is supported on port security (single VTEP) for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.

- Port-security is not supported with Fabric Peering.

- If the L2 Port Security feature is enabled, the following behavior is observed:

  - Secure MACs will be sent as static MACs and will be seen as static MACs on remote VTEPs. Hence, if there is any attempt to learn the secure MAC on a remote VTEP as a dynamic MAC (due to the malicious host with the same MAC), it will be prevented.

  - If **restrict** option is set on the violated MACs, then these violated MACs will be sent using the **static drop** set. On remote VTEPs, the MACs will be configured with the **static drop** so that any attempt to send traffic to these hosts from the remote VTEPs will be dropped at the remote VTEP itself.

  - Both the local static and secure MAC is advertised to fabric with a sticky bit, so for a remote VTEP there is no difference if the remote static MAC is from a VTEP for secure or static MAC.

  - If local static exists already, that will take precedence over the remote static (either it is from secure or static).

  - There might be multiple updates for a MAC learned on a secure port from local VTEP to fabric based on the security decision made locally for the MAC, however, the final security behavior for the MAC will be consistent for the local and remote VTEP.

  - You can specify the **inactivity** value for a secure MAC. If there is no activity, then the secure MAC will be removed, and the secure MAC host can move to another port.

# Verifying the Port Security Support for VXLAN EVPN

To display the Port Security support for VXLAN EVPN configuration information, enter one of the following commands:

| Command | Purpose |
|---------|---------|
| **show running-config interface** *<interface-name>* | Displays running configuration of interface. |
| **show port-security** | Displays port security configuration information. |

### Example of show running-config interface command

```
switch(config-if)# show run inter e1/48
!Command: show running-config interface Ethernet1/48
!Running configuration last done at: Thu Feb 16 08:39:43 2023
!Time: Fri Feb 17 06:07:33 2023

version 10.3(3) Bios:version 01.08

interface Ethernet1/48
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 200
  spanning-tree port type edge trunk
  switchport port-security maximum 1025
```

```
    switchport port-security
    no shutdown

LVTEP:
------
switch(config-if)# show mac address-table inter e1/48
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
    VLAN     MAC Address     Type      age     Secure NTFY Ports
---------+----------------+--------+---------+------+----+-----------------
*  101    0012.0100.0001   secure   -          T      F    Eth1/48
*  101    0012.0100.0002   secure   -          T      F    Eth1/48
*  101    0012.0100.0003   secure   -          T      F    Eth1/48
*  101    0012.0100.0004   secure   -          T      F    Eth1/48
```

### Example of show port-security command

```
switch(config-if)# show port-security
Total Secured Mac Addresses in System (excluding one mac per port)     : 1024
Max Addresses limit in System (excluding one mac per port) : 7168


--------------------------------------------------------------------------------
Secure Port     MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                  (Count)        (Count)        (Count)
--------------------------------------------------------------------------------
Ethernet1/48         1025           1025            0              Shutdown

switch(config-if)# show port-security address interface e1/48


-----------------------------------------------------------------------
                   Secure Mac Address Table
-----------------------------------------------------------------------
Vlan    Mac Address     Type      Remaining  Remotely Remotely Ports
                                  age        learnt   aged
                                  (mins)              out
----    -----------    ------    ------     -------  -----    ----
 101    0012.0100.0001  DYNAMIC   0           No       No      Ethernet1/48
 101    0012.0100.0002  DYNAMIC   0           No       No      Ethernet1/48
 101    0012.0100.0003  DYNAMIC   0           No       No      Ethernet1/48
 101    0012.0100.0004  DYNAMIC   0           No       No      Ethernet1/48


RVTEP:
------
Standalone_VTEP_EX# show mac address-table
C  101     0012.0100.0001   static   -          F      F     nve1(20:20:20::20)
C  101     0012.0100.0002   static   -          F      F     nve1(20:20:20::20)
C  101     0012.0100.0003   static   -          F      F     nve1(20:20:20::20)
C  101     0012.0100.0004   static   -          F      F     nve1(20:20:20::20)
```

# Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a Cisco NX-OS device.

This chapter includes the following sections:

# 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 can be enabled globally and on a per-VLAN basis. By default, the feature is disabled globally and on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

# Trusted and Untrusted Sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source may 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 device that is under your administrative control. These devices include the switches, routers, and servers in the network. Any device 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 device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Cisco NX-OS 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 devices (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, all DHCP servers must be connected to the device 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 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 device receives specific DHCP messages. For example, the feature adds an entry to the database when the device receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the device 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.

Dynamic ARP inspection (DAI) and IP Source Guard also use information stored in the DHCP snooping binding database.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

# DHCP Snooping in a vPC Environment

A virtual port channel (vPC) allows two Cisco NX-OS switches to appear as a single logical port channel to a third device. The third device can be a switch, a server, or any other networking device that supports port channels.

In a typical vPC environment, DHCP requests can reach one vPC peer switch, and the responses can reach the other vPC peer switch, resulting in a partial DHCP (IP-MAC) binding entry in one switch and no binding entry in the other switch. As a result, DHCP snooping and associated features such as dynamic ARP inspection (DAI) and IP Source Guard are disrupted. This issue is addressed by using Cisco Fabric Service over Ethernet (CFSoE) distribution to ensure that all DHCP packets (requests and responses) appear on both switches, which helps in creating and maintaining the same binding entry on both switches for all clients behind the vPC link.

CFSoE distribution also allows only one switch to forward the DHCP requests and responses on the vPC link. In non-vPC environments, both switches forward the DHCP packets.

## Synchronizing DHCP Snooping Binding Entries

The dynamic DHCP binding entries should be synchronized in the following scenarios:

- When the remote vPC is online, all the binding entries for that vPC link should be synchronized with the peer.

- When DHCP snooping is enabled on the peer switch, the dynamic binding entries for all vPC links should be synchronized with the peer.

# Packet Validation

The device validates DHCP packets received on the untrusted interfaces of VLANs that have DHCP snooping enabled. The device forwards the DHCP packet unless any of the following conditions occur (in which case, the packet is dropped):

- The device receives a DHCP response packet (such as a DHCPACK, DHCPNAK, or DHCPOFFER packet) on an untrusted interface.

- The device receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping MAC address verification option is turned on.

- The device receives a DHCPRELEASE or DHCPDECLINE message from an untrusted host with an entry in the DHCP snooping binding table, and the interface information in the binding table does not match the interface on which the message was received.

In addition, you can enable strict validation of DHCP packets, which checks the options field of DHCP packets, including the "magic cookie" value in the first four bytes of the options field. By default, strict validation is disabled. When you enable it, by using the **ip dhcp packet strict-validation** command, if DHCP snooping processes a packet that has an invalid options field, it drops the packet.

# DHCP Snooping Option 82 Data Insertion

DHCP can centrally manage the IP address assignments for a large number of subscribers. When you enable Option 82, the device identifies a subscriber device that connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can connect to the same port on the access device and are uniquely identified.

When you enable Option 82 on the Cisco NX-OS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.

2. When the Cisco NX-OS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier vlan-ifindex (for non-vPCs) or vlan-vpcid (for vPCs), from which the packet is received (the circuit ID suboption).

> **Note** For vPC peer switches, the remote ID suboption contains the vPC switch MAC address, which is unique in both switches. This MAC address is computed with the vPC domain ID. The Option 82 information is inserted at the switch where the DHCP request is first received before it is forwarded to the other vPC peer switch.

3. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.

4. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.

5. The DHCP server sends the reply to the Cisco NX-OS device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

If the previously described sequence of events occurs, the following values do not change:

- Circuit ID suboption fields

  - Suboption type

  - Length of the suboption type

  - Circuit ID type

  - Length of the circuit ID type

- Remote ID suboption fields

  - Suboption type

  - Length of the suboption type

  - Remote ID type

  - Length of the circuit ID type

This figure shows the packet formats for the remote ID suboption and the circuit ID suboption. The Cisco NX-OS device uses the packet formats when you globally enable DHCP snooping and when you enable Option 82 data insertion and removal. For the circuit ID suboption, the module field is the slot number of the module.

**Figure 9: Suboption Packet Formats**

**Circuit ID Suboption Frame Format**

| Suboption type | Length | Circuit ID type | Length | VLAN | ifindex/vpcid |
|---|---|---|---|---|---|
| 1 | 10 | 2/3 | 8 | VLAN | ifindex/vpcid |
| 1 byte | 1 byte | 1 byte | 1 byte | 4 bytes | 4 bytes |

**Remote ID Suboption Frame Format**

| Suboption type | Length | Remote ID type | Length | MAC address |
|---|---|---|---|---|
| 2 | 8 | 0 | 6 | MAC address |
| 1 byte | 1 byte | 1 byte | 1 byte | 6 bytes |

349427

*Figure 10: Suboption Packet Format (Non TLV format)*

# About the DHCP Relay Agent

## DHCP Relay Agent

You can configure the device to run a DHCP relay agent, which forwards DHCP packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCP packet) and, if configured, adds the relay agent information option (Option 82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing Option 82.

After you enable Option 82, the device uses the binary ifindex format by default. If needed, you can change the Option 82 setting to use an encoded string format instead.

**Note**  When the device relays a DHCP request that already includes Option 82 information, the device forwards the request with the original Option 82 information without altering it.

## DHCP Relay Agent Option 82

You can enable the device to insert and remove Option 82 information on DHCP packets that are forwarded by the relay agent.

**Figure 11: DHCP Relay Agent in a Metropolitan Ethernet Network**

This figure shows an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the device at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.



When you enable Option 82 for the DHCP relay agent on the Cisco NX-OS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.

2. When the Cisco NX-OS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier ifindex (for non-VXLAN VLANs) or vn-segment-id-mod-port (for VXLAN VLANs), from which the packet is received (the circuit ID suboption). In DHCP relay, the circuit ID is filled with the ifindex of the SVI or Layer 3 interface on which DHCP relay is configured.

3. The device adds the IP address of the relay agent to the DHCP packet.

4. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.

5. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.

6. The DHCP server unicasts the reply to the Cisco NX-OS device if the request was relayed to the server by the device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

This figure shows the packet formats for the circuit ID suboption and the remote ID suboption.

*Figure 12: Suboption Packet Formats*

# VRF Support for the DHCP Relay Agent

You can configure the DHCP relay agent to forward DHCP broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCP servers in a different VRF. By using a single DHCP server to provide DHCP support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF. For general information about VRFs, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

Enabling VRF support for the DHCP relay agent requires that you enable Option 82 for the DHCP relay agent.

If a DHCP request arrives on an interface that you have configured with a DHCP relay address and VRF information and the address of the DCHP server belongs to a network on an interface that is a member of a different VRF, the device inserts Option 82 information in the request and forwards it to the DHCP server in the server VRF. The Option 82 information includes the following:

**VPN identifier**

Name of the VRF that the interface that receives the DHCP request is a member of.

**Link selection**

Subnet address of the interface that receives the DHCP request. When DHCP smart relay is enabled, the link selection is filled with the subnet of the active giaddr.

**Server identifier override**

IP address of the interface that receives the DHCP request. When DHCP smart relay is enabled, the server identifier is filled with the active giaddr.

**Note**   The DHCP server must support the VPN identifier, link selection, and server identifier override options.

When the device receives the DHCP response message, it strips off the Option 82 information and forwards the response to the DHCP client in the client VRF.

# DHCP Smart Relay Agent

When the DHCP relay agent receives broadcast DHCP request packets from a host, it sets giaddr to the primary address of the inbound interface and forwards the packets to the server. The server allocates IP addresses from the giaddr subnet pool until the pool is exhausted and ignores further requests.

You can configure the DHCP smart relay agent to allocate IP addresses from the secondary IP address subnet pool if the first subnet pool is exhausted or the server ignores further requests. This enhancement is useful if the number of hosts is greater than the number of IP addresses in the pool or if multiple subnets are configured on an interface using secondary addresses.

# 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 DHCP support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF. For general information about VRFs, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

## IPv6 Availability for Delegated Prefix Through the v6 Relay Agent

DHCPv6 Prefix Delegation (DHCPv6-PD) feature is aimed at assigning complete subnets and other network and interface parameters from a DHCPv6-PD server to a DHCPv6-PD client. It is an extension to DHCPv6 relay agent as defined in RFC3633.

**Note**
- Prefix delegation does not interwork with First Hop Security (FHS).
- DHCPv6 does not support prefix delegation with relay chaining.

The relay agent forwards the network address requests received in solicit packets to DHCPv6 server using an IANA option. If the client requires a Prefix address as well, then it adds an IAPD option in the request. DHCPv6 server delegates the requested Prefix, if it is available in its pool.

If CLI is enabled, the DHCPv6-PD adds a static route on the Switch for Delegated Prefix so that the prefix is routable from the Switch. DHCPv6-PD binding will be created for each client along with IPv6 route created for delegated prefix.

The added static routes are distributed to neighbors through an OSPFv3 routing protocol.

**Note** Currently the DHCPv6-PD routes distributions are not supported for other routing protocols like IS-IS, BGP. For more information, refer to the "Configuring Redistribution" section of *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

## DHCPv6 Smart Relay Agent

When the DHCPv6 smart relay agent receives solicit packets from a host, it sets link address to the address of the inbound interface and forwards the packets to the server. The server allocates IP addresses from the link address subnet pool until the pool is exhausted and ignores further requests.

You can configure the DHCPv6 smart relay agent to allocate IP addresses from the secondary IP address subnet pool if the first subnet pool is exhausted or the server ignores further requests. This enhancement is useful if the number of hosts is greater than the number of IP addresses in the pool or if multiple subnets are

configured on an interface using secondary addresses. You can allocate IP address from any address subnet pool.

# Guidelines and Limitations for DHCPv6 Smart Relay

DHCPv6 Smart Relay has the following configuration guidelines and limitations:

- In a vPC environment it is recommended that the subnet of the Ipv6 address(s) of an interface should be same on both the switches.

- The number of hosts that use DHCPv6 smart relay at an instance is restricted to 10000.

- It is supported on cloud-based platforms.

# About DHCP Client

The DHCP client feature enables the configuration of an IPv4 or IPv6 address on an interface. Interfaces can include routed ports, the management port, and switch virtual interfaces (SVIs).

# Prerequisites for DHCP

DHCP has the following prerequisite:

- You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent.

# Guidelines and Limitations for DHCP

DHCP has the following configuration guidelines and limitations:

- For secure POAP, make sure that DHCP snooping is enabled and firewall rules are set to block unintended or malicious DHCP servers.

- Cisco Nexus 9000 Series switches do not support the relaying of bootp packets. However, the switches do support bootp packets that are Layer 2 switched.

- DHCP subnet broadcast is not supported.

- You must enable the insertion of Option 82 information for DHCP packets to support the highest DHCP snooping scale.

- Before you globally enable DHCP snooping on the device, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.

- DHCP snooping should not be followed by DHCP relay in the network (DHCP snooping does not work when the DHCP relay is configured on the same Cisco Nexus device).

- The DHCP snooping, DAI, and FHS are not supported on Cisco Nexus 9500 platform switches with -R/-RX/-R2 line cards.

- DHCP snooping is not supported on VXLAN VLANs.

- DHCP snooping supports multiple IP addresses with the same MAC address and VLAN in static binding entries.

- VXLAN supports DHCP relay when the DHCP server is reachable through a default VRF.

- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, make sure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts. When both DHCP snooping and DHCP relay are enabled on a VLAN and the SVI of that VLAN, DHCP relay takes precedence.

- If an ingress router ACL is configured on a Layer 3 interface that you are configuring with a DHCP server address, make sure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

- If you use DHCP relay where DHCP clients and servers are in different VRFs, use only one DHCP server within a VRF.

- Access-control list (ACL) statistics are not supported if the DHCP snooping feature is enabled.

- Make sure that the DHCP configuration is synchronized across the switches in a vPC link. Otherwise, a run-time error can occur, resulting in dropped packets.

- DHCP Smart Relay is limited to the first 100 IP addresses of the interface on which it is enabled.

- You must configure a helper address on the interface in order to use DHCP smart relay.

- If DHCP Smart Relay is enabled in a vPC environment, primary interface IP addresses should share a subnet between the peers. Secondary interface IP addresses should also share a subnet between the peers.

- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.

- DHCPv6-PD Routes will not be generated when a DHCPv6 client initiates a Rebind. Existing IAPD entries for the client will be refreshed, but not created. For IAPD route creation, a full Solicit, Advertise, Request, Reply must be seen by the DHCPv6 Relay agent.

- If you use DHCP relay on an unnumbered interface, you must configure the switch to insert option 82.

- DHCPv6 Prefix Delegation Routes are not generated when Option 14 **Rapid Commit** is present. A full Solicit, Advertise, Request, Reply sequence is needed to generate an IAPD route.

- The following guidelines and limitations apply to the DHCP client feature:

  - You can configure multiple SVIs, but each interface VLAN should be in a different subnet. The DHCP client feature cannot configure different IP addresses with the same subnet on different interface VLANs on the same device.

  - DHCP client and DHCP relay are not supported on the same switch.

  - DHCP client is not supported for Layer 3 subinterfaces.

  - DHCP client is supported on the Cisco Nexus 9300 Series switches and the Cisco Nexus 9500 Series switches.

  - DHCP client is not supported on Cisco Nexus 9500 platform switches with N9K-X9636C-R, N9K-X9636C-RX, N9K-X9636Q-R, and N9K-X96136YC-R line cards.

- Beginning with Cisco NX-OS Release 9.3(3), DHCP snooping and DHCP relay is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

- Beginning with Cisco NX-OS Release 10.3(1)F, DHCP relay is supported on Cisco Nexus X9836DM-A line card of the Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, DHCP relay is supported on Cisco Nexus X98900CD-A line card of Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, DHCP relay is supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

**Note** For DHCP configuration limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

# Default Settings for DHCP

This table lists the default settings for DHCP parameters.

**Table 38: Default DHCP Parameters**

| Parameters | Default |
|---|---|
| DHCP feature | Disabled |
| DHCP snooping | Disabled |
| DHCP snooping on VLANs | Disabled |
| DHCP snooping MAC address verification | Enabled |
| DHCP snooping Option 82 support | Disabled |
| DHCP snooping trust | Untrusted |
| DHCP relay agent | Enabled |
| DHCPv6 relay agent | Enabled |
| VRF support for the DHCP relay agent | Disabled |
| VRF support for the DHCPv6 relay agent | Disabled |
| DHCP Option 82 for relay agent | Disabled |
| DHCP smart relay agent | Disabled |
| DHCP server IP address | None |

# Configuring DHCP

## Minimum DHCP Configuration

**Procedure**

| | |
|---|---|
| **Step 1** | Enable the DHCP feature. |
| | When the DHCP feature is disabled, you cannot configure DHCP snooping. |
| **Step 2** | Enable DHCP snooping globally. |
| **Step 3** | Enable DHCP snooping on at least one VLAN. |
| | By default, DHCP snooping is disabled on all VLANs. |
| **Step 4** | Make sure that the DHCP server is connected to the device using a trusted interface. |
| **Step 5** | (Optional) Enable the DHCP relay agent. |
| **Step 6** | (Optional) If DHCP servers and clients are in different VRFs, do the following: |
| | a) Enable Option 82 for the DHCP relay agent. |
| | b) Enable VRF support for the DHCP relay agent. |
| **Step 7** | (Optional) Configure an interface with the IP address of the DHCP server. |

## Enabling or Disabling the DHCP Feature

You can enable or disable the DHCP feature on the device. By default, DHCP is disabled.

When the DHCP feature is disabled, you cannot configure the DHCP relay agent, DHCP snooping, or any of the features that depend on DHCP. In addition, all DHCP configuration is removed from the device.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>```switch# configure terminal``` <br>```switch(config)#``` | Enters global configuration mode. |
| **Step 2** | [**no**] **feature dhcp**<br><br>**Example:**<br>```switch(config)# feature dhcp``` | Enables the DHCP feature. The **no** option disables the DHCP feature and erases all DHCP configuration. |
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:** | Displays the DHCP configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# show running-config dhcp | |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Configuring DHCP Snooping

## Enabling or Disabling DHCP Snooping Globally

You can enable or disable DHCP snooping globally on the device.

### Before you begin

Make sure that you have enabled the DHCP feature.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp snooping**<br><br>**Example:**<br>switch(config)# ip dhcp snooping | Enables DHCP snooping globally. The **no** form of this command disables DHCP snooping. |
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>switch(config)# show running-config dhcp | Displays the DHCP configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

## Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs. By default, DHCP snooping is disabled on all VLANs.

**Before you begin**

Make sure that the DHCP feature is enabled.

> **Note**  If a VACL is configured on a VLAN that you are configuring with DHCP snooping, make sure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

**Procedure**

|        | **Command or Action**                                   | **Purpose**                                                                                                      |
|--------|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Step 1 | **configure terminal**                                  | Enters global configuration mode.                                                                                |
|        | **Example:**                                            |                                                                                                                  |
|        | `switch# configure terminal`<br>`switch(config)#`       |                                                                                                                  |
| Step 2 | [**no**] **ip dhcp snooping vlan** *vlan-list*          | Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** form of this command disables DHCP snooping on the VLANs specified. |
|        | **Example:**                                            |                                                                                                                  |
|        | `switch(config)# ip dhcp snooping vlan`<br>`100,200,250-252` |                                                                                                              |
| Step 3 | (Optional) **show running-config dhcp**                 | Displays the DHCP configuration.                                                                                 |
|        | **Example:**                                            |                                                                                                                  |
|        | `switch(config)# show running-config dhcp`              |                                                                                                                  |
| Step 4 | (Optional)  **copy running-config startup-config**      | Copies the running configuration to the startup configuration.                                                  |
|        | **Example:**                                            |                                                                                                                  |
|        | `switch(config)# copy running-config`<br>`startup-config` |                                                                                                                |

## Enabling or Disabling DHCP Snooping MAC Address Verification

You can enable or disable DHCP snooping MAC address verification. If the device receives a packet on an untrusted interface and the source MAC address and the DHCP client hardware address do not match, address verification causes the device to drop the packet. MAC address verification is enabled by default.

**Before you begin**

Make sure that the DHCP feature is enabled.

**Procedure**

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

| | Command or Action | Purpose |
|---|---|---|
| | `switch# config t`<br>`switch(config)#` | |
| Step 2 | [**no**] **ip dhcp snooping verify mac-address**<br>**Example:**<br>`switch(config)# ip dhcp snooping verify`<br>`mac-address` | Enables DHCP snooping MAC address verification. The **no** form of this command disables MAC address verification. |
| Step 3 | (Optional) **show running-config dhcp**<br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| Step 4 | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

## Enabling or Disabling Option 82 Data Insertion and Removal

You can enable or disable the insertion and removal of Option 82 information for DHCP packets forwarded without the use of the DHCP relay agent. By default, the device does not include Option 82 information in DHCP packets.

**Note**  DHCP relay agent support for Option 82 is configured separately.

**Note**  To support a higher DHCP pps scale, you must enable the insertion of Option 82 information for DHCP packets.

**Note**  You must add Option82 as specified in the format string in the command configuration.

- The length of the Option82 string increases based on the length of the format string.

- The circuit-id must include the ascii value of the format string.

**Before you begin**

Make sure that the DHCP feature is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp snooping information option**<br><br>**Example:**<br><br>`switch(config)# ip dhcp snooping`<br>`information option` | Enables the insertion and removal of Option 82 information for DHCP packets. The **no** form of this command disables the insertion and removal of Option 82 information. |
| **Step 3** | (Optional) [**no**] **ip dhcp option82 sub-option circuit-id** *format_type string format*<br><br>**Example:**<br><br>`switch(config)# ip dhcp snooping`<br>`sub-option circuit-id format-type string`<br>` format`<br><br>**Example:**<br><br>`switch(config)# ip dhcp snooping`<br>`sub-option circuit-id format-type string`<br>` format?`<br>`WORD Format string (Max Size 64)` | Configures Option 82 as follows:<br><br>• If you do not specify *format-type*, the *circuit-id* displays the incoming port, for example, *ethernet1/1*.<br><br>• If you specify format *\<word\>*, the *circuit-id* displays the specified word<br><br>• If you specify *%h* instead of *\<word\>*, the *circuit-id* displays the host name.<br><br>• If you specify *%p* instead of *\<word\>*, the *circuit-id* displays the port name.<br><br>• If you specify *%h:%p* instead of *\<word\>*, the *circuit-id* displays both host and port name.<br><br>**Note** The *no* option disables this behavior. |
| **Step 4** | (Optional) [**no**] **ip dhcp snooping sub-option format non-tlv**<br><br>**Example:**<br><br>`switch(config)# ip dhcp snooping`<br>`sub-option format non-tlv` | Removes subtype, and its length, from Circuit ID and Remote ID suboptions of Option 82 information. |
| **Step 5** | **interface** *interface slot/port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/2`<br>`switch(config-if)#` | Enters the interface configuration mode, where slot/port is the interface where you want to enable or disable snooping. |
| **Step 6** | (Optional) **ip dhcp option82 sub-option circuit-id**<br><br>**Example:**<br><br>`switch(config-if)# ip dhcp option82`<br>`sub-option circuit-id?`<br>`WORD Format string (Max Size 64)` | Configures Option 82 at the interface.<br><br>**Note** This command is not supported at SVI and Sub-Interface.<br><br>**Note** The *no* option disables this behavior |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | **Example:**<br>`switch(config-if)# ip dhcp option82`<br>`sub-option circuit-id test`<br>`switch(config-if)#` | |
| **Step 7** | **exit**<br><br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |
| **Step 8** | (Optional) **show ip dhcp option82 info interface** *intf_name* | Displays the DHCP configuration. It shows whether option82 is enabled or disabled on that interface and the transmitted packets for an interface that is option82 enabled. |
| **Step 9** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 10** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`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 default, strict validation of DHCP packets is disabled.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [no] **ip dhcp packet strict-validation**<br><br>**Example:**<br>`switch(config)# ip dhcp packet`<br>`strict-validation` | Enables the strict validation of DHCP packets. The **no** form of this command disables strict DHCP packet validation. |
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`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. By default, all interfaces are untrusted. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces

- Layer 2 port-channel interfaces

### Before you begin

Make sure that the DHCP feature is enabled.

Make sure that the interface is configured as a Layer 2 interface.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Do one of the following options:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-number*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | • Enters interface configuration mode, where *slot*/*port* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.<br><br>• Enters interface configuration mode, where *slot*/*port* 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**<br><br>**Example:**<br>`switch(config-if)# ip dhcp snooping trust` | Configures the interface as a trusted interface for DHCP snooping. The **no** form of this command configures the port as an untrusted interface. |
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config-if)# show running-config`<br>`dhcp` | Displays the DHCP configuration. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

## Enabling or Disabling DHCP Relay Trusted Port Functionality

You can enable or disable the DHCP relay trusted port functionality. By default, if the gateway address is set to all zeros in the DHCP packet and the relay information option is already present in the packet, the DHCP relay agent will not discard the packet. If the **ip dhcp relay information option trust** command is configured globally, the DHCP relay agent will discard the packet if the gateway address is set to all zeros.

### Before you begin

Make sure that the DHCP feature is enabled.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp relay information option trust**<br><br>**Example:**<br><br>`switch(config)# ip dhcp relay information`<br>`option trust` | Enables the DHCP relay trusted port functionality. The **no** form of this command disables this functionality. |
| **Step 3** | (Optional) **show ip dhcp relay**<br><br>**Example:**<br><br>`switch(config)# show ip dhcp relay` | Displays the DHCP relay configuration. |
| **Step 4** | (Optional) **show ip dhcp relay information trusted-sources**<br><br>**Example:**<br><br>`switch(config)# show ip dhcp relay`<br>`information trusted-sources` | Displays the DHCP relay trusted ports configuration. |
| **Step 5** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

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

## Configuring an Interface as a DHCP Relay Trusted or Untrusted Port

You can configure whether a Layer 3 interface is a DHCP relay trusted or untrusted interface. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

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

**Before you begin**

Make sure that the DHCP feature is enabled.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | **interface** [**ethernet** *slot/port*[*.number*] \| **port-channel** *channel-number*]<br><br>**Example:**<br><br>switch(config)# interface ethernet 2/1<br>switch(config-if)# | Enters interface configuration mode, where *slot/port* is the Layer 3 Ethernet interface that you want to configure as trusted or untrusted or *channel-number* is the Layer 3 port-channel interface that you want to configure as trusted or untrusted. |
| **Step 3** | [**no**] **ip dhcp relay information trusted**<br><br>**Example:**<br><br>switch(config-if)# ip dhcp relay information trusted | Configures the interface as a trusted interface for DHCP relay agent information. The **no** form of this command configures the port as an untrusted interface.<br><br>**Note** For any Layer 3 interface, if the interface is configured as trusted either through a global command or an interface-level command, the interface is considered as a trusted interface. Hence, when the trusted-port command is enabled at the global level, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 4** | (Optional) **show ip dhcp relay information trusted-sources**<br><br>**Example:**<br>`switch(config-if)# show ip dhcp relay information trusted-sources` | Displays the DHCP relay trusted ports configuration. |
| **Step 5** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config-if)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

## Configuring all Interfaces as Trusted or Untrusted

You can configure all Layer 3 interfaces as DHCP relay trusted or untrusted interfaces. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

- Layer 3 Ethernet interfaces and subinterfaces

- Layer 3 port-channel interfaces

When you enable the **ip dhcp relay information trust-all** command, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration.

### Before you begin

Make sure that the DHCP feature is enabled.

### Procedure

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp relay information trust-all**<br><br>**Example:**<br>`switch(config)# ip dhcp relay information trust-all` | Configures the interfaces as trusted sources of DHCP messages. The **no** form of this command configures the ports as untrusted interfaces. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 3** | (Optional) **show ip dhcp relay information trusted-sources**<br><br>**Example:**<br>`switch(config)# show ip dhcp relay`<br>`information trusted-sources` | Displays the DHCP relay trusted ports configuration. |
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`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** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp relay**<br><br>**Example:**<br>`switch(config)# ip dhcp relay` | Enables the DHCP relay agent. The **no** option disables the DHCP relay agent. |
| **Step 3** | (Optional) **show ip dhcp relay**<br><br>**Example:**<br>`switch(config)# show ip dhcp relay` | Displays the DHCP relay configuration. |
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Enabling or Disabling Option 82 for the DHCP Relay Agent

You can enable or disable the device to insert and remove Option 82 information on DHCP packets forwarded by the relay agent.

By default, the DHCP relay agent does not include Option 82 information in DHCP packets.

### Before you begin

Ensure that the DHCP feature is enabled.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# [**no**] **ip dhcp relay information option** | Enables the DHCP relay agent to insert and remove Option 82 information on the packets that it forwards. The Option 82 information is in binary ifindex format by default. The **no** option disables this behavior. |
| **Step 3** | (Optional) switch(config)# [**no**] **ip dhcp relay sub-option circuit-id customized** | Programs Option 82 with the VLAN + slot + port format. This command is applicable only for SVIs. The **no** option disables this behavior. |
| **Step 4** | (Optional) [**no**] **ip dhcp relay sub-option format non-tlv**<br><br>**Example:**<br>`switch(config)# ip dhcp relay sub-option`<br>` format non-tlv` | Removes subtype, and its length, from Circuit ID and Remote ID suboptions of Option 82 information. |
| **Step 5** | (Optional) switch(config)# [**no**] **ip dhcp relay sub-option circuit-id format-type string** | Configures Option 82 to use encoded string format instead of the default binary ifindex format. The **no** option disables this behavior.<br><br>For VLANs and SVIs:<br><br>• When this command and the **ip dhcp relay sub-option circuit-id customized** command are both configured, the **ip dhcp relay sub-option circuit-id format-type string** command is programmed. |

| | Command or Action | Purpose |
|---|---|---|
| | | • When the **ip dhcp relay sub-option circuit-id format-type string** command is removed, the **ip dhcp relay sub-option circuit-id customized** command is programmed. |
| | | • When both commands are removed, the ifindex is programmed. |
| | | For other interfaces, if the **ip dhcp relay sub-option circuit-id format-type string** command is configured, it is used. Otherwise, the default ifindex is programmed. |
| Step 6 | (Optional) switch(config)# **show ip dhcp relay** | Displays the DHCP relay configuration. |
| Step 7 | (Optional) switch(config)# **show running-config dhcp** | Displays the DHCP configuration. |
| | | **Note** In the output of this command, the **circuit-id** parameter value will be displayed in double quotes, even if configured with or without double quotes. |
| 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. |

# Enabling or Disabling VRF Support for the DHCP Relay Agent

You can configure the device to support the relaying of DHCP requests that arrive on an interface in one VRF to a DHCP server in a different VRF.

### Before you begin

You must enable Option 82 for the DHCP relay agent.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>```switch# configure terminal``` <br>```switch(config)#``` | Enters global configuration mode. |
| Step 2 | [**no**] **ip dhcp relay information option vpn**<br><br>**Example:**<br><br>```switch(config)# ip dhcp relay information``` <br>``` option vpn``` | Enables VRF support for the DHCP relay agent. The **no** option disables this behavior. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 3** | [**no**] **ip dhcp relay sub-option type cisco**<br><br>**Example:**<br><br>`switch(config)# ip dhcp relay sub-option type cisco` | Enables DHCP to use Cisco proprietary numbers 150, 152, and 151 when filling the link selection, server ID override, and VRF name/VPN ID relay agent Option 82 suboptions. The **no** option causes DHCP to use RFC numbers 5, 11, and 151 for the link selection, server ID override, and VRF name/VPN ID suboptions. |
| **Step 4** | (Optional) **show ip dhcp relay**<br><br>**Example:**<br><br>`switch(config)# show ip dhcp relay` | Displays the DHCP relay configuration. |
| **Step 5** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Disabling the Server Identifier Override Option

Beginning with Cisco NX-OS Release 9.3(3), you can disable the server identifier override option. This option is added by default in DHCP Option 82 packets for a DHCP relay VPN configuration or source interface configuration.

**Before you begin**

You must enable Option 82 for the DHCP relay agent.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp relay information option server-id-override-disable**<br><br>**Example:**<br><br>`switch(config)# ip dhcp relay information option server-id-override-disable` | Disables the server identifier override option in DHCP Option 82 packets.<br><br>**Note**  You can use the **no** form of this command to re-enable the server identifier override option. |

# Configuring DHCP Server Addresses on an Interface

You can configure DHCP server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCP server IP addresses specified. The relay agent forwards replies from all DHCP servers to the host that sent the request.

**Before you begin**

Ensure that the DHCP feature is enabled.

Ensure that the DHCP server is correctly configured.

Determine the IP address for each DHCP server that you want to configure on the interface.

If the DHCP server is in a different VRF than the interface, ensure that you have enabled VRF support.

> **Note**    If an ingress router ACL is configured on an interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Do one of the following options:<br><br>   • **interface ethernet** *slot*/*port*[*.number*]<br>   • **interface vlan** *vlan-id*<br>   • **interface port-channel** *channel-id*[*.subchannel-id*]<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#` | • Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.<br><br>  **Note**  Port-channel subinterfaces are supported only in Cisco NX-OS Releases 6.1(2)I3(3) and 6.1(2)I3(3a). They are not supported in Cisco NX-OS Release 9.2(1).<br><br>• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.<br><br>• Enters interface configuration mode, where *channel-id*  is the ID of the port channel that you want to configure with a DHCP |

| | **Command or Action** | **Purpose** |
|---|---|---|
| | | server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID. |
| Step 3 | **ip dhcp relay address** *IP-address* [**use-vrf** *vrf-name*]<br><br>**Example:**<br>`switch(config-if)# ip dhcp relay address 10.132.7.120 use-vrf red` | Configures an IP address for a DHCP server to which the relay agent forwards BOOTREQUEST packets received on this interface.<br><br>To configure more than one IP address, use the **ip dhcp relay address** command once per address. |
| Step 4 | (Optional) **show ip dhcp relay address**<br><br>**Example:**<br>`switch(config-if)# show ip dhcp relay address` | Displays all the configured DHCP server addresses. |
| Step 5 | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config-if)# show running-config dhcp` | Displays the DHCP configuration. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring the DHCP Relay Source Interface

You can configure the source interface for the DHCP relay agent. By default, the DHCP 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. When DHCP relay source interface is configured, the device adds the configured source interface IP address as giaddr to the DHCP packet if source interface VRF is same as that of DHCP server VRF. Otherwise, IP address of the interface through which the server is reachable, will be used as giaddr.

**Before you begin**

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

Ensure CLI dhcp relay information option and ip dhcp relay information option vpn are enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp relay source-interface** *interface*<br><br>**Example:**<br><br>`switch(config)# ip dhcp relay`<br>`source-interface loopback 2` | Configures the source interface for the DHCP relay agent.<br><br>**Note** The DHCP 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 ip dhcp relay** [**interface** *interface*]<br><br>**Example:**<br><br>`switch(config)# show ip dhcp relay` | Displays the DHCP relay configuration. |
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Enabling or Disabling DHCP Smart Relay Globally

You can enable or disable DHCP smart relay globally on the device.

**Before you begin**

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:** | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | switch# configure terminal<br>switch(config)# | |
| Step 2 | [no] ip dhcp smart-relay global<br><br>**Example:**<br>switch(config)# ip dhcp smart-relay global | Enables DHCP smart relay globally. The **no** form of this command disables DHCP smart relay. |
| Step 3 | (Optional) **show ip dhcp relay**<br><br>**Example:**<br>switch(config)# show ip dhcp relay | Displays the DHCP smart relay configuration. |
| Step 4 | (Optional) **show running-config dhcp**<br><br>**Example:**<br>switch(config)# show running-config dhcp | Displays the DHCP configuration. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Enabling or Disabling DHCP Smart Relay on a Layer 3 Interface

You can enable or disable DHCP smart relay on Layer 3 interfaces.

**Before you begin**

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 2 | **interface** *interface slot*/*port*<br><br>**Example:**<br>switch(config)# interface ethernet 2/3<br>switch(config-if)# | Enters interface configuration mode, where *slot*/*port* is the interface for which you want to enable or disable DHCP smart relay. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 3 | [**no**] **ip dhcp smart-relay**<br><br>**Example:**<br>`switch(config-if)# ip dhcp smart-relay` | Enables DHCP smart relay on the interface. The **no** form of this command disables DHCP smart relay on the interface. |
| Step 4 | **exit**<br><br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |
| Step 5 | **exit**<br><br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| Step 6 | (Optional) **show ip dhcp relay**<br><br>**Example:**<br>`switch# show ip dhcp relay` | Displays the DHCP smart relay configuration. |
| Step 7 | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch# show running-config dhcp` | Displays the DHCP configuration. |
| Step 8 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

## Configuring DHCP Relay Subnet-Selection

If an interface includes both, a primary and a secondary IP address, then by default the DHCP relay uses the primary subnet to request the IP address allocation from the server. You must enable DHCP smart relay if you want the DHCP relay to use the secondary IP address. With smart relay enabled, DHCP relay first requests the IP address in the primary subnet. If it fails to get the IP address in the primary subnet, it requests the IP address of the secondary subnet. The IP address of the secondary subnet is not chosen by default.

With the introduction of the DHCP relay subnet selection feature, you have an option to choose the IP address of either the primary or the secondary subnet based on your requirements. When you configure the DHCP relay subnet selection, the DHCP relayed packet includes the subnet that is used in subnet-selection for a source and relay agent. If there is a VPN or a source interface option, the option 82 link selection is updated with the configured subnet.

The DHCP Smart relay and the subnet-selection configuration are mutually exclusive at the interface level. If DHCP Smart relay is enabled globally and the subnet-selection is configured on the interface level, then the interface configuration takes precedence.

With the DHCP VPN or the source interface option, the DHCP server must use the option 82 link-selection to assign the IP address.

**Before you begin**

Ensure that the DHCP feature is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface vlan** *vlan-id*<br><br>**Example:**<br><br>`switch(config)#interface vlan 3`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | **ip dhcp relay subnet-selection** *ip address*<br><br>**Example:**<br><br>`switch(config-if)#ip dhcp relay`<br>`subnet-selection 20.20.21.1` | Configures the DHCP relay subnet-selection for the specified IP address. |

# Configuring DHCPv6

## Enabling or Disabling the DHCPv6 Relay Agent

You can enable or disable the DHCPv6 relay agent. By default, the DHCPv6 relay agent is enabled.

**Before you begin**

Ensure that the DHCP feature is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ipv6 dhcp relay**<br><br>**Example:**<br><br>`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*] | Displays the DHCPv6 relay configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>switch(config)# show ipv6 dhcp relay | |
| Step 4 | [**no**] **ipv6 dhcp relay prefix-delegation route**<br><br>**Example:**<br>switch(config)# ipv6 dhcp relay prefix-delegation route | Enables the IPv6 prefix route addition. The **no** option disables the prefix route addition. |
| Step 5 | (Optional) **show ipv6 dhcp relay prefix-delegation**<br><br>**Example:**<br>switch(config)# show ipv6 dhcp relay prefix-delegation | Displays the DHCPv6 IAPD entries (Client IP, interface, lease timer and delegated prefix). |
| Step 6 | (Optional) **show running-config dhcp**<br><br>**Example:**<br>switch(config)# show running-config dhcp | Displays the DHCP configuration. |
| Step 7 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# 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**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 2 | [**no**] **ipv6 dhcp relay option vpn**<br><br>**Example:**<br>switch(config)# ipv6 dhcp relay option vpn | Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior. |

| | Command or Action | Purpose |
|---|---|---|
| Step 3 | **[no] ipv6 dhcp relay option type cisco**<br><br>**Example:**<br><br>`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*]<br><br>**Example:**<br><br>`switch(config)# show ipv6 dhcp relay` | Displays the DHCPv6 relay configuration. |
| Step 5 | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Enabling or Disabling DHCPv6 Smart Relay Globally

You can enable or disable DHCPv6 smart relay globally on the device.

**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**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **[no] ipv6 dhcp smart-relay global**<br><br>**Example:**<br><br>`switch(config)# ipv6 dhcp smart-relay global` | Enables DHCPv6 smart relay globally. The **no** form of this command disables DHCPv6 smart relay. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 3** | (Optional) **show ipv6 dhcp relay** **Example:** switch(config)# show ipv6 dhcp relay | Displays the DHCPv6 smart relay configuration. |
| **Step 4** | (Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp | Displays the DHCPv6 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 DHCPv6 Smart Relay on a Layer 3 Interface

You can enable or disable DHCP smart relay on Layer 3 interfaces.

### 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** | **interface** *interface slot*/*port* **Example:** switch(config)# interface ethernet 2/3 switch(config-if)# | Enters interface configuration mode, where *slot*/*port* is the interface for which you want to enable or disable DHCPv6 smart relay. |
| **Step 3** | [**no**] **ipv6 dhcp smart-relay** **Example:** switch(config-if)# ipv6 dhcp smart-relay | Enables DHCPv6 smart relay on the interface. The **no** form of this command disables DHCPv6 smart relay on the interface. |
| **Step 4** | **exit** **Example:** switch(config-if)# exit switch(config)# | Exits interface configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 5** | **exit**<br><br>**Example:**<br><br>`switch(config)# exit`<br>`switch#` | Exits global configuration mode. |
| **Step 6** | (Optional) **show ipv6 dhcp relay**<br><br>**Example:**<br><br>`switch# show ipv6 dhcp relay` | Displays the DHCPv6 smart relay configuration. |
| **Step 7** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch# show running-config dhcp` | Displays the DHCPv6 configuration. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring DHCPv6 Server Addresses on an Interface

You can configure DHCPv6 server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCPv6 server IP addresses specified. The relay agent forwards replies from all DHCPv6 servers to the host that sent the request.

**Before you begin**

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 server is correctly configured.

Determine the IP address for each DHCPv6 server that you want to configure on the interface.

If the DHCPv6 server is in a different VRF than the interface, ensure that you have enabled VRF support.

**Note** If an ingress router ACL is configured on an interface that you are configuring with a DHCPv6 server address, ensure that the router ACL permits DHCP traffic between DHCPv6 servers and DHCP hosts.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 2** | Do one of the following options:<br><br>    • **interface ethernet** *slot*/*port*<br>    • **interface port-channel** *channel-id*<br><br>**Example:**<br>switch(config)# interface ethernet 2/3<br>switch(config-if)# | • Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCPv6 server IP address.<br><br>• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCPv6 server IP address. |
| **Step 3** | [**no**] **ipv6 dhcp relay address** *IPv6-address* [**use-vrf** *vrf-name*] [**interface** *interface*]<br><br>**Example:**<br>switch(config-if)# ipv6 dhcp relay address FF02:1::FF0E:8C6C use-vrf red | Configures an IP address for a DHCPv6 server to which the relay agent forwards BOOTREQUEST packets received on this interface.<br><br>Use the **use-vrf** option to specify the VRF name of the server if it is in a different VRF and the other argument interface is used to specify the output interface for the destination.<br><br>The server address can either be a link-scoped unicast or multicast address or a global or site-local unicast or multicast address. The **interface** option is mandatory for a link-scoped server address and multicast address. It is not allowed for a global or site-scoped server address.<br><br>To configure more than one IP address, use the **ipv6 dhcp relay address** command once per address. |
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>switch(config-if)# show running-config dhcp | Displays the DHCPv6 configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config-if)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Enabling DHCPv6 Option 79

Beginning with Cisco NX-OS Release 9.3(3), you can enable the use of the DHCPv6 client's link-layer address through Option 79. When you enable this feature, the switch adds Option 79 with relay forward packets, and the IPv6 client's link-layer address is inserted into the Options field of the DHCPv6 packet.

This feature is supported for both regular DHCPv6 and DHCPv6 with VXLAN.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | **ipv6 dhcp relay option79**<br><br>**Example:**<br><br>switch(config)# **ipv6 dhcp relay option79** | Enables the DHCP relay forward packets that are transmitted from the relay server to the DHCP server to carry the DHCPv6 host's link-layer address.<br><br>This command affects the transmitted relay forward packets only. |

# 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**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | [**no**] **ipv6 dhcp relay source-interface** *interface*<br><br>**Example:**<br><br>switch(config)# ipv6 dhcp relay source-interface loopback 2 | Configures the source interface for the DHCPv6 relay agent.<br><br>**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*]<br><br>**Example:** | Displays the DHCPv6 relay configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# show ipv6 dhcp relay` | |
| Step 4 | (Optional) **show running-config dhcp  show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP configuration. |
| Step 5 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring IPv6 RA Guard

You can configure the IPv6 router advertisement (RA) guard feature for Cisco Nexus 9200, 9300, and 9300-EX Series switches and the N9K-X9732C-EX line card. This feature is used to drop all incoming IPv6 RA packets on a Layer 2 interface.

### Before you begin

You must enable DHCP (using the **feature dhcp** command).

To enable DHCP relay on any interface, you must disable DHCP on interfaces that have an IPv4 or IPv6 address assigned using DHCP (dynamic IP addressing).

Make sure that both PTP (**feature ptp**) and NV overlay (**feature nv overlay**) are not already configured. A dynamic ifacl label is reserved when these features are configured. However, only two dynamic ifacl label bits are available. If both of these features are already configured, a dynamic ifacl label will not be available for IPv6 RA guard, and the feature cannot be enabled.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface** *interface slot/port*<br><br>**Example:**<br>`switch(config)# interface ethernet 2/2`<br>`switch(config-if)#` | Enters interface configuration mode. |
| Step 3 | [**no**] **ipv6 nd raguard**<br><br>**Example:**<br>`switch(config-if)# ipv6 nd raguard` | Enables the IPv6 RA guard feature on the specified interface. |

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

# Enabling DHCP Client

You can use the DHCP client feature to enable the configuration of an IPv4 or IPv6 address on an interface. Interfaces can include routed ports, the management port, and switch virtual interfaces (SVIs). Layer 3 subinterfaces are not supported.

**Note** DHCP client is independent of the DHCP relay and DHCP snooping processes, so it does not require that the **feature dhcp** command be enabled.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | Do one of the following options:<br><br>  • **interface ethernet** *slot*/*port*<br>  • **interface mgmt 0**<br>  • **interface vlan** *vlan-id*<br><br>**Example:**<br>`switch(config)# interface vlan 3`<br>`switch(config-if)#` | • Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface for which you want to enable the DHCP client feature.<br><br>• Enters interface configuration mode and specifies the management interface as the interface for which you want to enable the DHCP client feature.<br><br>• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN for which you want to enable the DHCP client feature. |
| **Step 3** | **ipv6 address use-link-local-only**<br><br>**Example:**<br>`switch(config-if)# ipv6 address`<br>`use-link-local-only` | You must enter this command before assigning an IPv6 address to the interface in the next step. This command is not required if you will assign an IPv4 address to the interface. |
| **Step 4** | [**no**] {**ip** | **ipv6**} **address dhcp** | Assigns an IPv4 or IPv6 address to the interface. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`switch(config-if)# ip address dhcp` | The **no** form of this command releases the IP address. |
| **Step 5** | (Optional) Do one of the following options:<br>• **show running-config interface ethernet** *slot*/*port*<br>• **show running-config interface mgmt 0**<br>• **show running-config interface vlan** *vlan-id*<br>**Example:**<br>`switch(config-if)# show running-config interface vlan 3` | Displays the IPv4 or IPv6 address assigned to the interface in the running configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration.<br><br>Only the {**ip** \| **ipv6**} **address dhcp** command is saved. The assigned IP address is not saved even though it shows in the running configuration. |

# Configuring UDP Relay

## About UDP Relay

By default, routers do not forward broadcast packets. You must configure routers if you want to forward broadcast packets. You can use the UDP relay feature to relay broadcasts destined for UDP ports except DHCPv4 port numbers 67 and 68. The UDP relay feature is also known as the IP Helper feature.

Us the **ip forward-protocol udp** command to enable the UDP relay feature. By default, the UDP relay feature is disabled.

To forward a packet, configure IP address object groups with the forwarding destination IP addresses or network addresses and then associate the IP address object groups with the L3 interfaces.

The UDP relay feature is supported on the following types of Layer 3 interfaces:

- Physical port
- Interface VLAN (SVI)
- L3 port channel
- L3 subinterfaces

## Guidelines and Limitations for UDP Relay

UDP relay has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(5), UDP relay is supported on Cisco Nexus 9200, 9332C, 9364C, 9300-EX, 9300-FX/FX2/FXP platform switches, and Cisco Nexus 9500 platform switches with -EX/FX line cards.

- The UDP relay is not supported on Cisco Nexus 9300-FX3/GX/GX2/H2R/H1 Series switches and Cisco Nexus 9500 platform switches with -R/-RX line cards.

- The UDP port must be in the range of 1 to 65565.

- Any L3 or SVI interface can be associated with a maximum of one object group. Therefore, any interface can be associated with a maximum of 300 UDP relay IP addresses.

- The UDP relay feature supports seven UDP ports.

- The objet-group name can be maximum of 64 alpha-numeric characters.

- DHCP and UDP relay cannot co-exist.

- Subnet broadcast is not supported.

# Configuring UDP Relay

### Before you begin

Ensure that you have enabled the DHCP feature.

### Procedure

**Step 1**    **configure terminal**

**Example:**
```
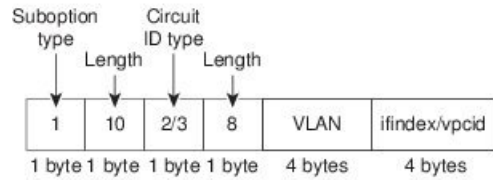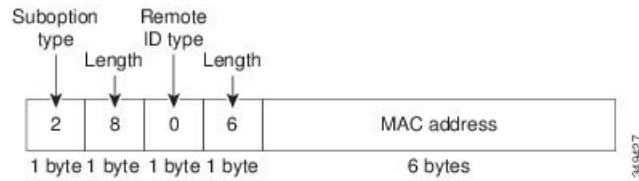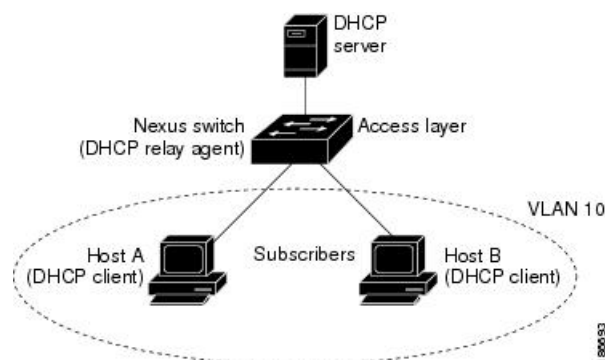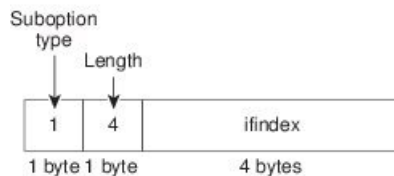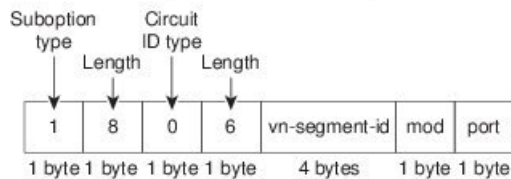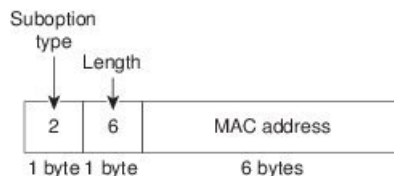switch# configure terminal
switch(config)#
```

Enters global configuration mode.

**Step 2**    **[no] ip forward-protocol udp**

**Example:**
```
switch(config)# ip forward-protocol udp
```

Enables the UDP relay feature. By default, the UDP relay feature is disabled. However, it is enabled on the predefined set of UDP ports.

**Step 3**    (Optional) **[no] ip forward-protocol udp** *udp-port-number*

**Example:**
```
switch(config)# ip forward-protocol udp 1
```

Enable the UDP relay feature on the non-default UDP ports.

**Note**    You can enable or disable UDP forwarding for any UDP port in the range 1 to 65565 except the DHCP ports.

**Step 4**    **[no] object-group udp relay ip addres** *object-group-name*

**Example:**

```
switch(config)# ip forward-protocol udp relay ip address relay1
```

Configures the destination IP addresses to which the packets are forwarded.

**Note**   For each entry that you want to create, use the **host** command to specify a single host, or omit the **host** command to specify a network of hosts.

**Step 5**   **[no] {host** *host-addr| network-addr network-mask| network-addr/mask-length*}

**Example:**

```
switch(config)# host 2.1.2.2 30.1.1.1 255.255.255.0 10.1.1.1./24
```

Configure an object group that consists of destination IP addresses to which the packets are forwarded

**Note**   For each entry that you want to create, use the **host** command to specify a single host, or omit the **host** command to specify a network of hosts.

**Step 6**   **exit**

**Example:**

```
switch(config-udp-group)# exit
```

Exists the interface configuration mode.

**Step 7**   **interface ethernet** *slot/port*

**Example:**

```
switch(config)# interface ethernet 1/1
```

Associates the object froup with a Layer 3 interface.

**Note**   The L3 interface can be a physical port, interface VLAN (SVI), L3 port channel, or L3 subinterfaces.

**Step 8**   **ip udp relay addrgroup** *object-group-name*

**Example:**

```
switch(config-if)# ip udp relay addrgroup group1
```

Associates an object group to the interface.

**Step 9**   **exit**

**Example:**

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

Exists the interface configuration mode.

# Configuration Example for UDP Relay

The following example shows a running configuration to configure UDP relay.

### Configuring UDP Relay

This example shows a running configuration to configure the UDP relay feature.

```
configure terminal
 feature dhcp
 ip forward-protocol udp
 object-group udp relay ip address <udprelay1>
   host <20.1.2.2>
   <30.1.1.1> <255.255.255.0>
   <10.1.1.1/24>
   exit
interface ethernet <e1/1>
 ip udp relay addrgroup <udprelay1>
 exit
```

# Verifying the UDP Relay Configuration

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

| Command | Purpose |
|---------|---------|
| **show ip udp relay** | Displays the UDP replay configuration. |
| **show ip udp relay interface** [{ *interface-type* \| *interface-range}]* | Displays the interface level attributes. |
| **show ip udp relay object-group** | Displays all configured UDP relay object-groups and the associated IP addresses. |
| **show ip udp relay object-group** *object-group-name* | Displays the object-group and the associated IP addresses. |

# Verifying the DHCP Configuration

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

| Command | Purpose |
|---------|---------|
| **show ip dhcp relay** | Displays the DHCP relay configuration. |
| **show ipv6 dhcp relay** [**interface** *interface*] | Displays the DHCPv6 relay global or interface-level configuration. |

| Command | Purpose |
|---------|---------|
| **show ipv6 dhcp relay prefix-delegation** | Displays the DHCPv6 IAPD entries on Relay agent with following options:<br><br>• **client**: Displays the prefix bindings for a client.<br><br>**detail**: Displays the detailed information.<br><br>**interface**: Displays the prefix bindings for an interface.<br><br>**prefix**: Displays a specific prefix binding. |
| **show ipv6 route dhcpv6** | Displays connected routes owned by dhcpv6 with following options:<br><br>• **all**: Displays the routes for protocol for backup next-hops too.<br><br>**bind-label**: Displays the routes with this bind-label only.<br><br>**detail**: Displays the routes in full detail<br><br>**interface**: Displays the routes with this output interface only<br><br>**next-hop**: Displays the routes with this next-hop only.<br><br>**summary**: Displays the route counts.<br><br>**updated**: Displays the routes filtered by last updated time.<br><br>**vrf**: Displays per-VRF information.<br><br>**Note**    Ensure that the DHCPv6-PD feature is enabled. |
| **show ip dhcp relay address** | Displays all the DHCP server addresses configured on the device. |
| **show ip dhcp snooping** | Displays general information about DHCP snooping. |

| Command | Purpose |
|---|---|
| **show running-config dhcp** [**all**] | Displays the DHCP configuration in the running configuration. |
| | **Note** The **show running-config dhcp** command displays the **ip dhcp relay** and the **ipv6 dhcp relay** commands, although these are configured by default. |
| **show running-config interface** {**ethernet** *slot/port* \| **mgmt 0** \| **vlan** *vlan-id*} | Displays the IPv4 or IPv6 address assigned to the interface when DHCP client is enabled. |
| show startup-config dhcp [**all**] | Displays the DHCP configuration in the startup configuration. |

# Displaying IPv6 RA Guard Statistics

To display IPv6 RA guard statistics, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show ipv6 raguard statistics** | Displays IPv6-related RA guard statistics. |

The following example shows sample statistics:

```
switch# show ipv6 raguard statistics
-----------------------------------
Interface     Rx         Drops
------------- ---------- ------------
Ethernet1/53  4561102    4561102
```

# Displaying DHCP Snooping Bindings

Use the **show ip dhcp snooping binding** [*ip-address* \| *mac-address* \| **dynamic** \| **static** \| **vlan** *vlan-id* \| **interface** *interface-type interface-number*]  command to display all entries from the DHCP snooping binding database.

```
MacAddress         IpAddress LeaseSec Type   VLAN Interface
----------------- --------- -------- ------ ---- ------------
0f:00:60:b3:23:33 10.3.2.2  infinite static 13   Ethernet2/46
0f:00:60:b3:23:35 10.2.2.2  infinite static 100  Ethernet2/10
```

# Clearing the DHCP Snooping Binding Database

Use the **clear ip dhcp snooping binding** command to clear all entries from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding interface ethernet** *slot/port* command to clear entries associated with a specific Ethernet interface from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding interface port-channel** *channel-number* command to clear entries associated with a specific port-channel interface from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding vlan** *vlan-id* [**mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot /port* | **port-channel** *channel-number*}] command to clear a single specific VLAN entry from the DHCP snooping binding database.

# Monitoring DHCP

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.

Use the **show ip dhcp relay statistics** [**interface** *interface*] command to monitor DHCP relay statistics at the global or interface level.

Use the **show ipv6 dhcp relay statistics** [**interface** *interface*] command to monitor DHCPv6 relay statistics at the global or interface level.

# Clearing DHCP Snooping Statistics

Use the **clear ip dhcp snooping statistics** [**vlan** *vlan-id*] command to clear the DHCP snooping statistics.

# 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 global statistics** command to clear the DHCP statistics globally.

# 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.

# Clearing DHCPv6-PD Binding

The following clear commands clears binding and corresponding IPv6 route as well.

In typical vPC environment, clearing DHCPv6-PD binding at vPC peer switch will clear binding along with corresponding IPv6 route at other vPC peer switch.

Use the **clear ipv6 dhcp relay prefix-delegation all** command to clear all entries in the DHCPv6-PD binding.

Use the **clear ipv6 dhcp relay prefix-delegation client** command to clear Client's IPv6 address entries in the DHCPv6-PD binding.

Use the **clear ipv6 dhcp relay prefix-delegation interface** command to clear entries associated with a specific interface in the DHCPv6-PD binding.

**Note**    Post valid PD entry clearing through CLI, no further PD entry and route learning with Renew/Rebind packet.

# Configuration Examples for DHCP

This 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 information option

interface ethernet 2/5
  ip dhcp snooping trust
ip dhcp snooping vlan 1
ip dhcp snooping vlan 50
```

This example shows how to enable the DHCP relay agent and configure the DHCP server IP address for Ethernet interface 2/3, where the DHCP server IP address is 10.132.7.120 and the DHCP server is in the VRF instance named red:

```
feature dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn

interface ethernet 2/3
  ip dhcp relay address 10.132.7.120 use-vrf red
```

This example shows how to enable and use the DHCP smart relay agent. In this example, the device forwards the DHCP broadcast packets received on Ethernet interface 2/2 to the DHCP server (10.55.11.3), inserting 192.168.100.1 in the giaddr field. If the DHCP server has a pool configured for the 192.168.100.0/24 network, it responds. If the server does not respond, the device sends two more requests using 192.168.100.1 in the giaddr field. If the device still does not receive a response, it starts using 172.16.31.254 in the giaddr field instead.

```
feature dhcp
ip dhcp relay
ip dhcp smart-relay global

interface ethernet 2/2
  ip address 192.168.100.1/24
  ip address 172.16.31.254/24 secondary
  ip dhcp relay address 10.55.11.3
```

# Configuration Examples for DHCP Client

The following example shows how the DHCP client feature can be used to assign an IPv4 address to a VLAN interface:

```
switch# configure terminal
switch(config)# interface vlan 7
switch(config-if)# no shutdown
switch(config-if)# ip address dhcp
switch(config-if)# show running-config interface vlan 7
interface Vlan7
no shutdown
ip address dhcp
```

# Additional References for DHCP

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Dynamic ARP inspection (DAI) | *Cisco Nexus 9000 Series NX-OS Security Configuration Guide* |
| IP Source Guard | *Cisco Nexus 9000 Series NX-OS Security Configuration Guide* |
| vPCs | *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* |
| VRFs and Layer 3 virtualization | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

**Standards**

| Standards | Title |
|---|---|
| RFC 2131 | Dynamic Host Configuration Protocol (https://datatracker.ietf.org/doc/html/rfc2131) |
| RFC 3046 | DHCP Relay Agent Information Option (https://datatracker.ietf.org/doc/html/rfc3046) |
| RFC3633 | IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 (RFC) (https://tools.ietf.org/html/rfc3633) |
| RFC 6607 | Virtual Subnet Selection Options for DHCPv4 and DHCPv6 (https://datatracker.ietf.org/doc/html/rfc6607) |
| RFC 6939 | Client Link-Layer Address Option in DHCPv6 (https://datatracker.ietf.org/doc/html/rfc6939) |

C H A P T E R **17**

# Configuring IPv6 First Hop Security

This chapter describes how to configure First Hop Security (FHS) features on Cisco NX-OS devices.

This chapter includes the following sections:

## About First-Hop Security

The Layer 2 and Layer 3 switches operate in the Layer 2 domains with technologies such as server virtualization, Overlay Transport Virtualization (OTV), and Layer 2 mobility. These devices are sometimes referred to as "first hops", specifically when they are facing end nodes. The First-Hop Security feature provides end node protection and optimizes link operations on IPv6 or dual-stack networks.

First-Hop Security (FHS) is a set of features to optimize IPv6 link operation, and help with scale in large L2 domains. These features provide protection from a wide host of rogue or mis-configured users. You can use extended FHS features for different deployment scenarios, or attack vectors.

The following FHS features are supported:

- IPv6 RA Guard

- DHCPv6 Guard

- IPv6 Snooping

✎

**Note**     See Guidelines and Limitations of First-Hop Security, on page 472 for information about enabling this feature.

**Note**  Use the **feature dhcp** command to enable the FHS features on a switch.

# IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 snooping, DHCPv6 guard, and IPv6 RA guard are IPv6 global policies features. Each time IPv6 snooping, DHCPv6 guard, or RA guard is configured globally, the policy attributes are stored in the software policy database. The policy is then applied to an interface, and the software policy database entry is updated to include this interface to which the policy is applied.

Use the **hardware access-list tcam region ing-redirect tcam_size** command, to configure FHS. You can resize the **ing-racl** region to allocate space to the **ing-redirect** region.

- Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches, FHS packets take the copp-s-dhcpreq queue for software processing.

- Cisco Nexus 9300, 9500 platform switches, N9K-X9432C-S line card use the class default.

**Note**  When you upgrade the Cisco Nexus Series switch to Cisco NX-OS Release 7.0(3)I7(1) using the In-Service Software Upgrades (ISSU), you must reload the Cisco NX-OS box before configuring the port level FHS policies.

# IPv6 First-Hop Security Binding Table

A database table of IPv6 neighbors connected to the device is created from information sources such as IPv6 snooping. This database, or binding table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

# Guidelines and Limitations of First-Hop Security

The general guidelines and limitations of First-Hop Security are as follows:

- Before enabling the FHS on the interface or VLAN, we recommend carving TCAM regions on Cisco Nexus 9300 and 9500 Series switches. To enable FHS successfully:

  - On an interface, you must carve the **ifacl** TCAM region.

  - On a VLAN, you must carve the necessary redirect TCAM region.

  - On a FEX interface, you must carve the **fex-ipv6-ifacl** TCAM region.

- Before enabling the FHS, we recommend carving the **ing-redirect** TCAM region onCisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches.

- Beginning with Cisco NX-OS Release 9.3(5), FHS is supported on Cisco Nexus 9300-GX switches.

# About vPC First-Hop Security Configuration

You can deploy IPv6 First-Hop Security vPC in many ways. We recommend the following best practice deployment scenarios:

- DHCP relay on-stack

- DHCP relay on vPC leg

- DHCP client and relay on orphan ports

## DHCP Relay On-stack

In this deployment scenario, you can directly connect clients behind the vPC link, or behind an intermediary switch with DHCP relay running on the Nexus switch. Connecting clients behind an intermediary switch with DHCP relay running on the Nexus switch, is ideal because you can configure the IPv6 Snooping feature on the vPC interface links directly, instead of at a VLAN level. Configuration at the interface level is efficient for the following reasons:

- Control traffic (DHCP/ND) will not be redirected to CPU for processing on both vPC peers if it goes over the peer link.

- Packets switched over the peer link aren't processed a second time.

*Figure 13: FHS Configuration with DHCP relay on-stack*



In the figure, snooping policy is enabled on both vPC links. In this scenario, the two vPC peers learn all the host IP/MAC bindings behind the vPC links and sync these up between themselves. The two vPC peers learn the bindings using both IPv6 ND and IPv6 DHCP control protocols.

# DHCP Relay on VPC Leg

In this configuration, the relay agent does not run on the vPC peers. Instead, the DHCP relay agent (or a DHCP server) is runs behind a vPC link (it can be towards the access, or even somewhere in the core). In such a deployment scenario, the IPv6 Snooping feature doesn't implicitly trust the DHCP Server messages and drops DHCP Server messages by default. You can customize the IPv6 policy to implement:

- Security-level glean.

- IPv6 DHCP Guard policy with device-role server. In this configuration, IPv6 Snooping trusts DHCP server messages attached to the vPC link.

*Figure 14: FHS Configuration with external DHCP relay*



In the figure, the clients are located behind the vPC links with the default IPv6 snooping policy. You can attach both ipv6 snooping and ipv6 dhcp-guard attach-policy SERVER policies to the links where DHCP server traffic arrives. You will need both the server or relay facing and client facing IPv6 snooping policies to create the client binding entries via DHCP control traffic. This is because IPv6 Snooping needs to see both the client and server packets to create the binding. You must also configure the IPv6 DHCP Guard policy to allow DHCP server traffic by the IPv6 Snooping policy. Both peers require the same configuration because the vPC peers synch all newly learnt client entries learnt on the vPC port.

# DHCP Client Relay on Orphan Ports

In this configuration, you can connect the client via an orphan port. The IPv6 Snooping feature only syncs client bindings on vPC ports, but not on orphan ports as these are not directly connected to both vPC peers. In such a configuration, the IPv6 Snooping feature runs independently on both switches. The figure illustrates the following:

- On the first switch, you must attach the IPv6 Snooping policy on the client facing interface. However, to accommodate DHCP server packets coming from the server on an orphan port behind the vPC peer, you must attach the policy at the VLAN level. In such a case, the policy applied at the VLAN inspects both the client traffic interface and DHCP server traffic. You do not require an individual IPv6 snooping policy per interface. Any DHCP traffic arriving via the vPC peer is also implicitly trusted and if policing is required, the vPC peer automatically drops it.

- You must also configure IPv6 on the second switch at the VLAN level. You must also configure the IPv6 DHCP Guard policy with a "device-role server" on the server facing orphan port. This prevents the IPv6 Snooping feature from dropping the DHCP server packets. Both switches learn the client binding entries individually and will not sync them, because the client is not on a vPC link.

*Figure 15: FHS configuration with client and DHCP relay on orphan port*



# RA Guard

## Overview of IPv6 RA Guard

The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The IPv6 RA Guard feature analyzes these RAs and filters out RAs that are sent by unauthorized devices. In host mode, all RA and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 (L2) device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

# IPv6 RA Router Advertisement and the Flags

The Router Advertisement suggests to devices how to create or obtain a global unicast address and other addressing information for communicating on the link. The RA message uses four flags to tell devices how this is to be done:

1. Address Autoconfiguration flag (A flag): The A flag is enabled by default. This flag tells to hosts on the local link that the specified prefix can be used for IPv6 autoconfiguration.

2. Other Configuration flag (O flag): The O flag is disabled by default. This flag tells the host to get addressing information other than its global unicast address from a stateless DHCPv6 server. This information may include DNS server addresses and a domain name.

3. Managed Address Configuration flag (M flag): The M flag is disabled by default. This flag tells a host to use a stateful DHCPv6 server for its global unicast address and all other addressing information. When stateful DHCPv6 is required, use the **ipv6 managed-config-flag** command to enable the M Flag.

> **Note** When the M flag is enabled, the A flag should usually be disabled. Manually enabling the M flag does not automatically disable the A flag. To disable the A flag, use the **ipv6 nd prefix** *ipv6-prefix/prefix-length* **no-autoconfig** command.

4. On-Link flag (L flag): The L flag is also enabled by default. The L flag identifies that a specific prefix is on this link or subnet. IPv6 does not perform the Logical AND hashing to determine whether a destination IP address is local to the link as IPv4 does. If the L flag is disabled, every packet is sent to the default gateway. The A flag and the L flag are advertised via ICMPv6 Router Advertisement (RA) by default.

# Guidelines and Limitations of IPv6 RA Guard

The guidelines and limitations of IPv6 RA Guard are as follows:

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.

- Beginning with Cisco NX-OS Release 10.1(1), IPV6 RA guard is supported on Cisco Nexus 9300-GX platform switches.

- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.

- This feature can be configured on a switch port interface in the ingress direction.

- This feature supports host mode and router mode.

- This feature is supported only in the ingress direction; it is not supported in the egress direction.

- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.

- Packets dropped by the IPv6 RA Guard feature can be spanned.

# DHCPv6 Guard

## Overview of DHCP—DHCPv6 Guard

The DHCPv6 Guard feature blocks DHCP reply and advertisement messages that originate from unauthorized DHCP servers and relay agents that forward DHCP packets from servers to clients. Client messages or messages sent by relay agents from clients to servers are not blocked. The filtering decision is determined by the device role assigned to the receiving switch port, trunk, or VLAN. This functionality helps to prevent traffic redirection or denial of service (DoS).

Packets are classified into one of the three DHCP type messages. All client messages are always switched regardless of device role. DHCP server messages are only processed further if the device role is set to server. Further processing of DHCP server advertisements occurs for server preference checking.

If the device is configured as a DHCP server, all the messages need to be switched, regardless of the device role configuration.

## Limitation of DHCPv6 Guard

The guidelines and limitations of DHCPv6 Guard are as follows:

- If a packet arriving from DHCP server is a Relay Forward or a Relay Reply, only the device role is checked. In addition, IPv6 DHCP Guard doesn't apply the policy for a packet sent out by the local relay agent running on the switch.

# IPv6 Snooping

## Overview of IPv6 Snooping

IPv6 "snooping," feature bundles several Layer 2 IPv6 first-hop security features, which operates at Layer 2, or between Layer 2 and Layer 3, and provides IPv6 features with security and scalability. This feature mitigates some of the inherent vulnerabilities for the neighbor discovery mechanism, such as attacks on duplicate address detection (DAD), address resolution, device discovery, and the neighbor cache.

IPv6 snooping learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables and analyzes snooping messages in order to build a trusted binding table. IPv6 snooping messages that do not have valid bindings are dropped. An IPv6 snooping message is considered trustworthy if its IPv6-to-MAC mapping is verifiable.

When IPv6 snooping is configured on a target (which varies depending on platform target support and may include device ports, switch ports, Layer 2 interfaces, Layer 3 interfaces, and VLANs), capture instructions are downloaded to the hardware to redirect the snooping protocol and Dynamic Host Configuration Protocol (DHCP) for IPv6 traffic up to the switch integrated security features (SISF) infrastructure in the routing device. For snooping traffic, Neighbor Discovery Protocol (NDP) messages are directed to SISF. For DHCPv6, UDP messages sourced from dhcvp6_client and dhcvp_server ports are redirected.

IPv6 snooping registers its "capture rules" to the classifier, which aggregates all rules from all features on a given target and installs the corresponding ACL down into the platform-dependent modules. Upon receiving

redirected traffic, the classifier calls all entry points from any registered feature (for the target on which the traffic is being received), including the IPv6 snooping entry point. This entry point is the last to be called, so any decision (such as drop) made by another feature supersedes the IPv6 snooping decision.

IPv6 snooping provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

Additionally, IPv6 snooping is the foundation for many other IPv6 features that depend on an accurate binding table. It inspects snooping and DHCP messages on a link to glean addresses, and then populates the binding table with these addresses. This feature also enforces address ownership and limits the number of addresses any given node is allowed to claim.

# Guidelines and Limitations for IPv6 Snooping

The guidelines and limitations of IPv6 Snooping are as follows:

- You must perform the same configurations on both the vPC peers. Automatic consistency checker for IPv6 snooping is not supported.

- The IPv6 Snooping feature is supported only in hardware when the ternary content addressable memory (TCAM) is programed.

- This feature can be configured on a switch port interface or VLAN only on the ingress port.

- For IPv6 Snooping to learn DHCP bindings, it must see both server and client replies. A IPv6 snooping policy must be attached to both the client facing the interface (or VLAN) as well as the DHCP server facing interface (or VLAN). In the case of DHCP Relay, an IPv6 Snooping policy must be attached at the VLAN level to see the server replies.

# How to Configure IPv6 FHS

## Configuring the IPv6 RA Guard Policy on the Device

**Note**    When the **ipv6 nd raguard** command is configured on ports, router solicitation messages are not replicated to these ports. To replicate router solicitation messages, all ports that face routers must be set to the router role.

**Procedure**

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

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 2** | **ipv6 nd raguard policy** *policy-name*<br><br>**Example:**<br><br>`Device(config)# ipv6 nd raguard policy policy1` | Defines the RA guard policy name and enters RA guard policy configuration mode. |
| **Step 3** | **device-role** {**host** \| **router** \| **monitor** \| **switch**}<br><br>**Example:**<br><br>`Device(config-raguard-policy)# device-role router` | Specifies the role of the device attached to the port.<br><br>• device-role host—Interface or VLAN where you connect a regular node or host. This where you apply the IPV6 RA Guard policy. The device-role host allows incoming RS packets, and blocks incoming RA or RR packets. RS packets that are received on another interface, are not redirected to the device-role host. Only RA and RR packets (that are allowed) are redirected to the device-role host.<br><br>• device-role switch—The device-role switch behaves similar to the device-role host. For example, you can use it as a label for a trunk port.<br><br>• device-role monitor—This device monitors network traffic. It behaves similar to the device-role host, except that RS packets are also sent to this interface. This helps capture traffic.<br><br>• device-role router—Interface that connects to the router. This interface allows incoming RS, RA, or RR packets. |
| **Step 4** | **hop-limit** {**maximum** \| **minimum** *limit*}<br><br>**Example:**<br><br>`Device(config-raguard-policy)# hop-limit minimum 3` | (Optional) Enables verification of the advertised hop count limit.<br><br>• If not configured, this check will be bypassed. |
| **Step 5** | **managed-config-flag** {**on** \| **off**}<br><br>**Example:**<br><br>`Device(config-raguard-policy)# managed-config-flag on` | (Optional) Enables verification that the advertised managed address configuration flag is on.<br><br>**Note**   When enabling the M flag, it is recommended to disable the A flag.<br><br>• If not configured, this check will be bypassed. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **other-config-flag** {**on** | **off**}<br><br>**Example:**<br><br>`Device(config-raguard-policy)#`<br>`other-config-flag on` | (Optional) Enables verification of the advertised "other" configuration parameter. |
| **Step 7** | **router-preference maximum** {**high** | **low** | **medium**}<br><br>**Example:**<br><br>`Device(config-raguard-policy)#`<br>`router-preference maximum high` | (Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit. |
| **Step 8** | **trusted-port**<br><br>**Example:**<br><br>`Device(config-raguard-policy)#`<br>`trusted-port` | (Optional) Specifies that this policy is being applied to trusted ports.<br><br>• All RA guard policing will be disabled. |
| **Step 9** | **exit**<br><br>**Example:**<br><br>`Device(config-raguard-policy)# exit` | Exits RA guard policy configuration mode and returns to global configuration mode. |

## Configuring IPv6 RA Guard on an Interface

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *type* *number*<br><br>**Example:**<br><br>`Device(config)# interface ethernet 1/1`<br><br>**Example:**<br><br>`Device(config)# vlan configuration 10` | Specifies an interface type and number, and places the device in interface or VLAN configuration mode. |
| **Step 3** | **ipv6 nd raguard attach-policy** [*policy-name*]<br><br>**Example:**<br><br>`Device(config-if)# ipv6 nd raguard`<br>`attach-policy` | Applies the IPv6 RA Guard feature to a specified interface. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 4** | **exit**<br><br>**Example:**<br><br>`Device(config-if)# exit` | Exits interface configuration mode. |
| **Step 5** | **show ipv6 nd raguard policy** [*policy-name*]<br><br>**Example:**<br><br>`switch# show ipv6 nd raguard policy host`<br>`Policy host configuration:`<br>`  device-role host`<br><br>`Policy applied on the following`<br>`interfaces:`<br><br>`  Et0/0      vlan all`<br>`  Et1/0      vlan all` | Displays the RA guard policy on all interfaces configured with the RA guard. |
| **Step 6** | **debug ipv6 snooping raguard** [*filter* \| *interface* \| *vlanid*]<br><br>**Example:**<br><br>`Device# debug ipv6 snooping raguard` | Enables debugging for IPv6 RA guard snooping information. |

# Configuring DHCP—DHCPv6 Guard

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **ipv6 dhcp guard policy** *policy-name*<br><br>**Example:**<br><br>`Device(config)# ipv6 dhcp guard policy`<br>` pol1` | Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode. |
| **Step 3** | **device-role** {**client** \| **server**}<br><br>**Example:**<br><br>`Device(config-dhcpg-policy)# device-role`<br>` server` | Specifies the device role of the device attached to the target (interface or VLAN).<br><br>• device-role client—Interface where a normal DHCPv6 client is connected. It blocks any incoming server packets.<br><br>• device-role server—Interface where a normal DHCPv6 server is connected. It |

| | Command or Action | Purpose |
|---|---|---|
| | | allows all DHCPv6 packets originating on this interface. |
| Step 4 | **preference min** *limit*<br><br>**Example:**<br><br>Device(config-dhcpg-policy)# preference min 0 | (Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed. |
| Step 5 | **preference max** *limit*<br><br>**Example:**<br><br>Device(config-dhcpg-policy)# preference max 255 | (Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed. |
| Step 6 | **trusted-port**<br><br>**Example:**<br><br>Device(config-dhcpg-policy)# trusted-port | (Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled. |
| Step 7 | **exit**<br><br>**Example:**<br><br>Device(config-dhcpg-policy)# exit | Exits DHCP guard configuration mode and returns to global configuration mode. |
| Step 8 | **interface** *type number*<br><br>**Example:**<br><br>Device(config)# interface GigabitEthernet 0/2/0 | Specifies an interface and enters interface configuration mode. |
| Step 9 | **switchport**<br><br>**Example:**<br><br>Device(config-if)# switchport | Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration. |
| Step 10 | **ipv6 dhcp guard** [**attach-policy** *policy-name*]<br><br>**Example:**<br><br>Device(config-if)# ipv6 dhcp guard attach-policy pol1 | Attaches a DHCPv6 guard policy to an interface. |
| Step 11 | **exit**<br><br>**Example:**<br><br>Device(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 12** | **vlan configuration** *vlan-id*<br><br>**Example:**<br><br>`Device(config)# vlan configuration 1` | Specifies a VLAN and enters VLAN configuration mode. |
| **Step 13** | **ipv6 dhcp guard** [**attach-policy** *policy-name*]<br><br>**Example:**<br><br>`Device(config-vlan-config)# ipv6 dhcp guard attach-policy pol1` | Attaches a DHCPv6 guard policy to a VLAN. |
| **Step 14** | **exit**<br><br>**Example:**<br><br>`Device(config-vlan-config)# exit` | Exits VLAN configuration mode and returns to global configuration mode. |
| **Step 15** | **exit**<br><br>**Example:**<br><br>`Device(config)# exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Step 16** | **show ipv6 dhcp guard policy** [*policy-name*]<br><br>**Example:**<br><br>`Device# show ipv6 dhcp policy guard pol1` | (Optional) Displays the policy configuration as well as all the interfaces where the policy is applied. |

# Configuring IPv6 Snooping

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **ipv6 snooping policy** *policy-name*<br><br>**Example:**<br>`Device(config)# ipv6 snooping policy policy1` | Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode. |
| **Step 3** | **device-role** { **node** | **switch** }<br><br>**Example:**<br>`Device(config-snoop-policy)# device-node switch` | Specifies the role of the device attached to the target (interface or VLAN):<br><br>• node—is the default. Bindings are created and entries are probed. |

| | Command or Action | Purpose |
|---|---|---|
| | | • switch—Entries are not probed and when a trusted port is enabled, bindings are not created. |
| **Step 4** | **[no] limit address-count**<br><br>**Example:**<br><br>Device(config-snoop-policy)# limit address-count 500 | Limits the number of binding entries, a no limit address-count means no limit. |
| **Step 5** | **[no] protocol** *dhcp* \| *ndp*<br><br>**Example:**<br><br>Device(config-snoop-policy)# protocol dhcp<br><br>Device(config-snoop-policy)# protocol ndp | Turns on or switches off either DHCP or NDP gleaning. |
| **Step 6** | **trusted-port**<br><br>**Example:**<br><br>Device(config-snoop-policy)# trusted-port | Specifies that the policy be applied to a trusted port. If an entry is a trusted-port, none of it's traffic will be blocked or dropped. |
| **Step 7** | **security-level** *glean* \| *guard* \| *inspect*<br><br>**Example:**<br><br>Device(config-snoop-policy)# security-level guard | Specifies the type of security applied to the policy: glean, guard, or inspect. Here is what each security level means:<br><br>• glean—learns bindings but does not drop packets.<br><br>• inspect—learns bindings and drops packets in case it detects an issue, such as address theft.<br><br>• guard—works like inspect, but in addition drops IPv6, ND, RA, and IPv6 DHCP Server packets in case of a threat. |
| **Step 8** | **tracking**<br><br>**Example:**<br><br>Device(config-snoop-policy)# tracking enable | Enables tracking. |
| **Step 9** | **exit**<br><br>**Example:**<br><br>Device(config-snoop-policy)# exit | Exits snooping configuration mode and returns to global configuration mode. |
| **Step 10** | **interface** *type-number*<br><br>**Example:** | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config-if)# interface ethernet 1/25` | |
| Step 11 | **[no] switchport**<br><br>Example:<br>`Device(config-if)# switchport` | Switches between Layer 2 and Layer 3 mode. |
| Step 12 | **ipv6 snooping attach-policy** *policy-name*<br><br>Example:<br>`Device(config-if)# ipv6 snooping attach-policy policy1` | Attaches the IPv6 snooping policy to an interface. |
| Step 13 | **exit**<br><br>Example:<br>`Device(config-if)# exit` | Exits interface configuration mode and returns to global configuration mode. |
| Step 14 | **vlan configuration** *vlan-id*<br><br>Example:<br>`Device(config)# vlan configuration 333` | Specifies a VLAN and enters VLAN configuration mode. |
| Step 15 | **ipv6 snooping attach-policy** *policy-name*<br><br>Example:<br>`Device(config-vlan-config)# ipv6 snooping attach-policy policy1` | Attaches the IPv6 snooping policy to a VLAN. |
| Step 16 | **exit**<br><br>Example:<br>`Device(config-vlan-config)# exit` | Exits VLAN configuration mode and returns to global configuration mode. |
| Step 17 | **exit**<br><br>Example:<br>`Device(config)# exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| Step 18 | **show ipv6 snooping policy** *policy-name*<br><br>Example:<br>`Device(config)# show ipv6 snooping policy policy1` | Displays the policy configuration and the interfaces where the policy is applied. |

## Verifying and Troubleshooting IPv6 Snooping

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **show ipv6 snooping capture-policy** [**interface** *type number*] | Displays snooping message capture policies. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br><br>Device# show ipv6 snooping capture-policy<br> interface ethernet 0/0 | |
| Step 2 | **show ipv6 snooping counter** [**interface** *type number*]<br>**Example:**<br><br>Device# show ipv6 snooping counter<br>interface FastEthernet 4/12 | Displays information about the packets counted by the interface counter. |
| Step 3 | **show ipv6 snooping features**<br>**Example:**<br><br>Device# show ipv6 snooping features | Displays information about snooping features configured on the device. |
| Step 4 | **show ipv6 snooping policies** [**interface** *type number*]<br>**Example:**<br><br>Device# show ipv6 snooping policies | Displays information about the configured policies and the interfaces to which they are attached. |
| Step 5 | **debug ipv6 snooping**<br>**Example:**<br><br>Device# debug ipv6 snooping | Enables debugging for snooping information in IPv6. |

# Configuration Examples

## Example: IPv6 RA Guard Configuration

```
Device(config)# interface ethernet 1/1

Device(config-if)# ipv6 nd raguard attach-policy

Device# show running-config interface ethernet 1/1

Building configuration...
Current configuration : 129 bytes
!
interface ethernet1/1
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
```

```
 ipv6 nd raguard
 end
```

# Example: Configuring DHCP—DHCPv6 Guard

The following example displays a sample configuration for DHCPv6 Guard:

```
configure terminal
ipv6 dhcp guard policy pol1
 device-role server
 preference min 0
 preference max 255
 trusted-port
interface GigabitEthernet 0/2/0
 switchport
 ipv6 dhcp guard attach-policy pol1
 vlan configuration 1
  ipv6 dhcp guard attach-policy pol1
show ipv6 dhcp guard policy pol1
```

# Example: Configuring IPv6 First-Hop Security Binding Table

```
config terminal
 ipv6 neighbor binding vlan 100 2001:db8::1 interface ethernet3/0
 ipv6 neighbor binding max-entries 100
 ipv6 neighbor binding logging
 ipv6 neighbor binding retry-interval 8
 exit
show ipv6 neighbor binding
```

# Example: Configuring IPv6 Snooping

```
switch (config)# ipv6 snooping policy policy1
switch(config-ipv6-snooping)# ipv6 snooping attach-policy policy1
switch(config-ipv6-snooping)# exit
.
.
.
Device# show ipv6 snooping policies policy1
Policy policy1 configuration:
  trusted-port
  device-role node
Policy applied on the following interfaces:
  Et0/0       vlan all
  Et1/0       vlan all
Policy applied on the following vlans:
  vlan 1-100,200,300-400
```

# Additional References for IPv6 First-Hop Security

This section includes additional information related to configuring IPv6 First-Hop Security.

**Related Documents**

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

**C H A P T E R  18**

# Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on a Cisco NX-OS device.

This chapter includes the following sections:

## About DAI

### ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

### ARP Spoofing Attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your Layer 2 network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic that is intended for other hosts on the subnet.

*Figure 16: ARP Cache Poisoning*

This figure shows an example of ARP cache poisoning.



Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address that is associated with IP address of IB. When host B receives the ARP request, the ARP cache on host B is populated with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds and the response reaches host A, the ARP cache on host A is populated with an ARP binding for a host with the IP address IB and MAC address MB. The device in between does not populate the ARP cache as both the request and the response are not destined to its local IP address.

Host C can poison the ARP caches of host A and host B by broadcasting two forged ARP responses with bindings: one for a host with the IP address of IA, a MAC address of MC, and another for a host with an IP address of IB and a MAC address of MC. Host B then uses the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Similarly, host A uses MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a *man-in-the middle* attack.

# DAI and ARP Spoofing Attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco Nexus device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports

- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination

- Drops invalid ARP packets

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database can also contain static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header.

# Interface Trust States and Network Security

DAI associates a trust state with each interface on the device. Packets that arrive on trusted interfaces bypass all DAI validation checks, and packets that arrive on untrusted interfaces go through the DAI validation process.

In a typical network configuration, the guidelines for configuring the trust state of interfaces are as follows:

**Untrusted**

Interfaces that are connected to hosts

**Trusted**

Interfaces that are connected to devices

With this configuration, all ARP packets that enter the network from a device bypass the security check. No other validation is needed at any other place in the VLAN or in the network.

⚠️

**Caution**    Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

*Figure 17: ARP Packet Validation on a VLAN Enabled for DAI*

The following figure assumes that both device A and device B are running DAI on the VLAN that includes host 1 and host 2. If host 1 and host 2 acquire their IP addresses from the DHCP server connected to device A, only device A binds the IP-to-MAC address of host 1. If the interface between device A and device B is untrusted, the ARP packets from host 1 are dropped by device B and connectivity between host 1 and host 2 is lost.



If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though device B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a device that runs DAI do not poison the ARP caches of other hosts in the network; however, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a device that runs DAI.

If some devices in a VLAN run DAI and other devices do not, the guidelines for configuring the trust state of interfaces on a device that runs DAI become the following:

**Untrusted**

Interfaces that are connected to hosts or to devices that are not running DAI

**Trusted**

Interfaces that are connected to devices that are running DAI

When you cannot determine the bindings of packets from devices that do not run DAI, isolate at Layer 3 the devices that run DAI from devices that do not run DAI.

**Note** Depending on your network setup, you may not be able to validate a given ARP packet on all devices in the VLAN.

# Logging DAI Packets

Cisco NX-OS maintains a buffer of log entries about DAI packets processed. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You can also specify the type of packets that are logged. By default, a Cisco Nexus device logs only packets that DAI drops.

If the log buffer overflows, the device overwrites the oldest DAI log entries with newer entries. You can configure the maximum number of entries in the buffer.

**Note** Cisco NX-OS does not generate system messages about DAI packets that are logged.

# DHCP Relay with Dynamic ARP Inspection

DAI uses DHCP snooping client binding database to validate the ARP packets. In releases earlier than Cisco NX-OS Release 10.1(1), this database was built by the DHCP Snooping process, which runs on the switch. The binding database isn't built when the switch acts as a DHCP relay. When snooping, DHCP relay and DAI are enabled together, the relay process takes precedence over snooping for processing incoming DHCP packets. Hence, snooping doesn't build the binding database. Since DAI depends on the binding database, it can't operate with DHCP relay. However, from Cisco NX-OS Release 10.1(1), you can build the binding database using DHCP relay DAI.

When a switch receives a DHCP request, a temporary binding entry is created consisting of the client's MAC address, VLAN, and the incoming interface. After receiving DHCPACK from the server, the binding entry is qualified. The offered IP address is added to the qualified temporary entry and the binding entry type is updated as dhcp-relay.

When you upgrade to Cisco NX-OS Release 10.1(1) or a later release and if you enable this feature, the ISSU proceeds without any error. Disable this feature before you downgrade from Cisco NX-OS Release 10.1(1) to an earlier release.

# Prerequisites for DAI

- You must enable the DHCP feature before you can configure DAI. See Configuring DHCP, on page 421.

- You must configure the VLANs on which you want to enable DAI. See the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*.

- You must configure the ACL TCAM region size for DAI using the **hardware access-list tcam region arp-ether** command. The DAI configuration will not be accepted unless the arp-ether region is effective. See Configuring ACL TCAM Region Sizes, on page 324.

# Guidelines and Limitations for DAI

DAI has the following configuration guidelines and limitations:

- DAI is an ingress security feature; it does not perform any egress checking.

- DAI is not effective for hosts connected to devices that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, you should separate the domain with DAI from domains without DAI. This separation secures the ARP caches of hosts in the domain with DAI.

- When you use the **feature dhcp** command to enable the DHCP feature, there is a delay of approximately 30 seconds before the I/O modules receive the DHCP or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with the DHCP feature disabled to a configuration with the DHCP feature enabled. For example, if you use the rollback feature to revert to a configuration that enables the DHCP feature, the I/O modules receive the DHCP and DAI configuration approximately 30 seconds after you complete the rollback.

- DAI is supported on access ports, trunk ports, and port-channel ports.

- The DAI trust configuration of a port channel determines the trust state of all physical ports that you assign to the port channel. For example, if you have configured a physical port as a trusted interface and then you add that physical port to a port channel that is an untrusted interface, the physical port becomes untrusted.

- When you remove a physical port from a port channel, the physical port does not retain the DAI trust state configuration of the port channel.

- When you change the trust state on the port channel, the device configures a new trust state on all the physical ports that comprise the channel.

- If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, make sure that you have configured the static IP-MAC address bindings.

- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, make sure that DHCP snooping is enabled.

- ARP ACLs are not supported.

- Beginning with Cisco NX-OS Release 9.3(3), DAI is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

# Guidelines and Limitations for DHCP Relay with DAI

- The following Cisco Nexus platform switches support this feature:

  - Cisco Nexus 9200 platform switches

  - Cisco Nexus 9300-EX platform switches

  - Cisco Nexus 9300-FX platform switches

- The binding database entries aren't stored in the hardware.

- The binding database is common for all VRFs. If there are multiple VRFs, map each VRF to a unique VLAN.

- IP Source Guard (IPSG) doesn't support this feature.

- Only IPv4 entries are stored in the binding database. IPv6 isn't supported.

- This feature doesn't support vPC.

# Default Settings for DAI

This table lists the default settings for DAI parameters.

**Table 39: Default DAI Parameters**

| Parameters | Default |
|---|---|
| DAI | Disabled on all VLANs. |
| Interface trust state | All interfaces are untrusted. |
| Validation checks | No checks are performed. |
| Log buffer | When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second. |
| Per-VLAN logging | All denied or dropped ARP packets are logged. |

# Configuring DAI

## Enabling or Disabling DAI on VLANs

You can enable or disable DAI on VLANs. By default, DAI is disabled on all VLANs.

**Before you begin**

Make sure that the DHCP feature is enabled.

Make sure that the VLANs on which you want to enable DAI are configured.

Make sure that the ACL TCAM region size for DAI (arp-ether) is configured.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| **Step 2** | [**no**] **ip arp inspection vlan** *vlan-list*<br><br>**Example:**<br><br>switch(config)# ip arp inspection vlan 13 | Enables DAI for the specified list of VLANs. The **no** option disables DAI for the specified VLANs. |
| **Step 3** | (Optional) **show ip arp inspection vlan** *vlan-id*<br><br>**Example:**<br><br>switch(config)# show ip arp inspection vlan 13 | Displays the DAI configuration for a specific VLAN. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Configuring the DAI Trust State of a Layer 2 Interface

You can configure the DAI interface trust state of a Layer 2 interface. By default, all interfaces are untrusted.

A device forwards ARP packets that it receives on a trusted Layer 2 interface but does not check them.

On untrusted interfaces, the device intercepts all ARP requests and responses and verifies that the intercepted packets have valid IP-MAC address bindings before updating the local cache and forwarding the packet to the appropriate destination. If the device determines that packets have invalid bindings, it drops the packets and logs them according to the logging configuration.

**Before you begin**

If you are enabling DAI, make sure that the DHCP feature is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface** *type port/slot*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/1`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | [**no**] **ip arp inspection trust**<br><br>**Example:**<br><br>`switch(config-if)# ip arp inspection`<br>`trust` | Configures the interface as a trusted ARP interface. The **no** option configures the interface as an untrusted ARP interface. |
| **Step 4** | (Optional) **show ip arp inspection interface** *type port/slot*<br><br>**Example:**<br><br>`switch(config-if)# show ip arp inspection`<br>` interface ethernet 2/1` | Displays the trust state and the ARP packet rate for the specified interface. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Enabling or Disabling Additional Validation

You can enable or disable additional validation of ARP packets. By default, no additional validation of ARP packets is enabled. When no additional validation is configured, the source MAC address and the source IP address check against the IP-to-MAC binding entry for ARP packets is performed by using the ARP sender MAC address and the ARP sender IP address.

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can use the following keywords with the **ip arp inspection validate** command to implement additional validations:

**dst-mac**

Checks the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

**ip**

> Checks the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

**src-mac**

> Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, follow these guidelines:

- You must specify at least one of the keywords. You can specify one, two, or all three keywords.

- Each **ip arp inspection validate** command that you enter replaces the configuration from any previous commands. If you enter an **ip arp inspection validate** command to enable src-mac and dst-mac validations, and a second **ip arp inspection validate** command to enable ip validation, the src-mac and dst-mac validations are disabled when you enter the second command.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}<br><br>**Example:**<br><br>`switch(config)# ip arp inspection`<br>`validate src-mac dst-mac ip` | Enables additional DAI validation. The **no** form of this command disables additional DAI validation. |
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP snooping configuration, including the DAI configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring the DAI Logging Buffer Size

You can configure the DAI logging buffer size. The default buffer size is 32 messages.

## Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **[no] ip arp inspection log-buffer entries** *number*<br><br>**Example:**<br><br>`switch(config)# ip arp inspection`<br>`log-buffer entries 64` | Configures the DAI logging buffer size. The **no** option reverts to the default buffer size, which is 32 messages. The buffer size can be between 1 and 1024 messages. |
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:**<br><br>`switch(config)# show running-config dhcp` | Displays the DHCP snooping configuration, including the DAI configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring DAI Log Filtering

You can configure how the device determines whether to log a DAI packet. By default, the device logs DAI packets that are dropped.

## Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **[no] ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** \| **none** \| **permit**}<br><br>**Example:**<br><br>`switch(config)# ip arp inspection vlan`<br>`100 dhcp-bindings permit` | Configures DAI log filtering, as follows. The **no** form of this command removes DAI log filtering.<br><br>• **all**—Logs all packets that match DHCP bindings.<br><br>• **none**—Does not log packets that match DHCP bindings.<br><br>• **permit**—Logs packets permitted by DHCP bindings. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 3** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config)# show running-config dhcp` | Displays the DHCP snooping configuration, including the DAI configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Enabling DHCP Relay with DAI

You can create the binding database when DHCP relay and DAI are enabled. This feature is disabled by default.

### Before you begin

Enable DAI and DHCP relay. Enable DHCP snooping globally and on VLAN. See the *Configuring DHCP* chapter for more information.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **ip dhcp relay dai**<br><br>**Example:**<br>`switch(config)# ip dhcp relay dai` | Enables creation of binding database in the relay. |
| **Step 3** | (Optional) **show ip dhcp snooping binding relay**<br><br>**Example:**<br>`switch(config)# show ip dhcp snooping binding relay` | Displays the binding entries of the dhcp-relay type. |
| **Step 4** | (Optional) **show system internal dhcp database global config**<br><br>**Example:**<br>`switch(config)# show system internal dhcp database global config` | Displays if the relay DAI feature is enabled or not. |

# Verifying the DAI Configuration

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

| Command | Purpose |
| --- | --- |
| **show ip arp inspection** | Displays the status of DAI. |
| **show ip arp inspection interfaces** [**ethernet** *slot/port* | **port-channel** *number*] | Displays the trust state and ARP packet rate for a specific interface or port channel. |
| **show ip arp inspection log** | Displays the DAI log configuration. |
| **show ip arp inspection vlan** *vlan-id* | Displays the DAI configuration for a specific VLAN. |
| **show running-config dhcp** [**all**] | Displays the DAI configuration. |

# Monitoring and Clearing DAI Statistics

To monitor and clear DAI statistics, use the commands in this table.

| Command | Purpose |
| --- | --- |
| **show ip arp inspection statistics** [**vlan** *vlan-id*] | Displays DAI statistics. |
| **clear ip arp inspection statistics vlan** *vlan-id* | Clears DAI statistics. |
| **clear ip arp inspection log** | Clears DAI logs. |

# Configuration Examples for DAI

## Two Devices Support DAI

These procedures show how to configure DAI when two devices support DAI.

**Figure 18: Two Devices Supporting DAI**

The following figure shows the network configuration for this example. Host 1 is connected to device A, and Host 2 is connected to device B. Both devices are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to device A. Both hosts acquire their IP addresses from the same DHCP server. Device A has the bindings for Host 1 and Host 2, and device B has the binding for Host 2. Device A Ethernet interface 2/3 is connected to device B Ethernet interface 1/4.

DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically-assigned IP addresses.

- This configuration does not work if the DHCP server is moved from device A to a different location.

- To ensure that this configuration does not compromise security, configure Ethernet interface 2/3 on device A and Ethernet interface 1/4 on device B as trusted.

## Configuring Device A

To enable DAI and configure Ethernet interface 2/3 on device A as trusted, follow these steps:

**Procedure**

**Step 1**      While logged into device A, verify the connection between device A and device B.

```
switchA# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID           Local Intrfce   Hldtme  Capability  Platform      Port ID
switchB             Ethernet2/3     177     R S I       WS-C2960-24TC Ethernet1/4
switchA#
```

**Step 2**      Enable DAI on VLAN 1 and verify the configuration.

```
switchA# configure terminal
switchA(config)# ip arp inspection vlan 1
switchA(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration  : Enabled
Operation State : Active
switchA(config)#
```

**Step 3**      Configure Ethernet interface 2/3 as trusted.

```
switchA(config)# interface ethernet 2/3
switchA(config-if)# ip arp inspection trust
switchA(config-if)# exit
switchA(config)# exit
switchA# show ip arp inspection interface ethernet 2/3
 Interface      Trust State    Rate (pps)    Burst Interval
 ------------   ----------     ----------    --------------
 Ethernet2/3    Trusted           15             5
```

**Step 4** Verify the bindings.

```
switchA# show ip dhcp snooping binding
MacAddress        IpAddress        LeaseSec   Type           VLAN  Interface
----------------- ---------------  --------   ------------   ----  ------------
00:60:0b:00:12:89 10.0.0.1         0          dhcp-snooping  1     Ethernet2/3
switchA#
```

**Step 5** Check the statistics before and after DAI processes any packets.

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

If host 1 sends out two ARP requests with an IP address of 10.0.0.1 and a MAC address of 0002.0002.0002, both requests are permitted and are shown as follows:

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
```

If host 1 tries to send an ARP request with an IP address of 10.0.0.3, the packet is dropped, and an error message is logged.

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Ethernet2/3, vlan
1.([0002.0002.0002/10.0.0.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Fri Jan 23 2015])
```

The statistics display as follows:

```
switchA# show ip arp inspection statistics vlan 1
switchA#
Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 2
ARP Res Dropped    = 0
DHCP Drops         = 2
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

## Configuring Device B

To enable DAI and configure Ethernet interface 1/4 on device B as trusted, follow these steps:

### Procedure

**Step 1**  While logged into device B, verify the connection between device B and device A.

```
switchB# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID          Local Intrfce  Hldtme  Capability  Platform      Port ID
switchA            Ethernet1/4    120     R S I       WS-C2960-24TC Ethernet2/3
switchB#
```

**Step 2**  Enable DAI on VLAN 1 and verify the configuration.

```
switchB# configure terminal
switchB(config)# ip arp inspection vlan 1
switchB(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchB(config)#
```

**Step 3**  Configure Ethernet interface 1/4 as trusted.

```
switchB(config)# interface ethernet 1/4
switchB(config-if)# ip arp inspection trust
switchB(config-if)# exit
switchB(config)# exit
switchB# show ip arp inspection interface ethernet 1/4
 Interface       Trust State    Rate (pps)    Burst Interval
```

```
-------------    ----------    ----------    --------------
 Ethernet1/4     Trusted       15            5
switchB#
```

**Step 4**     Verify the list of DHCP snooping bindings.

```
switchB# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type          VLAN  Interface
-----------------  ---------------  --------  ------------  ----  -------------
00:01:00:01:00:01  10.0.0.2         4995      dhcp-snooping  1    Ethernet1/4
switchB#
```

**Step 5**     Check the statistics before and after DAI processes any packets.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 sends out an ARP request with the IP address 10.0.0.2 and the MAC address 0001.0001.0001, the packet is forwarded, and the statistics are updated.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 attempts to send an ARP request with the IP address 10.0.0.1, DAI drops the request and logs the following system message:

```
00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Ethernet1/4, vlan
1.([0001.0001.0001/10.0.0.1/0000.0000.0000/0.0.0.0/01:53:21 UTC Fri Jan 23 2015])
```

The statistics display as follows:

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
```

```
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 1
ARP Res Dropped    = 0
DHCP Drops         = 1
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

# Examples for DHCP Relay with DAI

The following example displays if the DHCP relay DAI feature is enabled or not. If the feature isn't enabled the value of the **DHCP Relay DAI enabled** entry in the database is **No**.

```
switch(config)# show system internal dhcp database global config

Snooping enabled: Yes
Snoop option-82 enabled: No
Relay enabled: Yes
.
.
DHCP Relay DAI enabled : No
Validate source mac: No
Validate destination mac: No
```

# Additional References for DAI

## Related Documents

| Related Topic | Document Title |
|---|---|
| ACL TCAM regions | Configuring IP ACLs, on page 289 |
| DHCP and DHCP snooping | Configuring DHCP, on page 421 |

## Standards

| Standard | Title |
|---|---|
| RFC-826 | An Ethernet Address Resolution Protocol (https://datatracker.ietf.org/doc/html/rfc826) |

# Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on Cisco NX-OS devices.

This chapter includes the following sections:

## About IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches one of two sources of IP and MAC address bindings:

- Entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table
- Static IP source entries that you configure

Filtering on trusted IP and MAC address bindings helps prevent spoofing attacks, in which an attacker uses the IP address of a valid host to gain unauthorized network access. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

You can enable IP Source Guard on Layer 2 interfaces that are not trusted by DHCP snooping. IP Source Guard supports interfaces that are configured to operate in access mode and trunk mode. When you initially enable IP Source Guard, all inbound IP traffic on the interface is blocked except for the following:

- DHCP packets, which DHCP snooping inspects and then forwards or drops, depending upon the results of inspecting the packet
- IP traffic from static IP source entries that you have configured on the Cisco NX-OS device

The device permits the IP traffic when DHCP snooping adds a binding table entry for the IP address and MAC address of an IP packet or when you have configured a static IP source entry.

The device drops IP packets when the IP address and MAC address of the packet do not have a binding table entry or a static IP source entry. For example, assume that the **show ip dhcp snooping binding** command displays the following binding table entry:

```
MacAddress          IpAddress    LeaseSec   Type            VLAN    Interface
-----------------   ----------   ---------  -------------   ----    ---------
00:02:B3:3F:3B:99   10.5.5.2     6943       dhcp-snooping   10      Ethernet2/3
```

If the device receives an IP packet with an IP address of 10.5.5.2, IP Source Guard forwards the packet only if the MAC address of the packet is 00:02:B3:3F:3B:99.

# Prerequisites for IP Source Guard

IP Source Guard has the following prerequisites:

- You must enable the DHCP feature and DHCP snooping before you can configure IP Source Guard. See Configuring DHCP, on page 421.

- You must configure the ACL TCAM region size for IP Source Guard using the **hardware access-list tcam region ipsg** command. See  Configuring ACL TCAM Region Sizes, on page 324.

**Note**   By default the ipsg region size is zero. You need to allocate enough entries to this region for storing and enforcing the SMAC-IP bindings.

# Guidelines and Limitations for IP Source Guard

IP Source Guard has the following configuration guidelines and limitations:

- IP Source Guard limits IP traffic on an interface to only those sources that have an IP-MAC address binding table entry or static IP source entry. When you first enable IP Source Guard on an interface, you may experience disruption in IP traffic until the hosts on the interface receive a new IP address from a DHCP server.

- IP Source Guard is dependent upon DHCP snooping to build and maintain the IP-MAC address binding table or upon manual maintenance of static IP source entries.

- IP Source Guard is not supported on fabric extender (FEX) ports or generic expansion module (GEM) ports.

- The following guidelines and limitations apply to the Cisco Nexus 9200 Series switches:

  - IPv6 adjacency is not formed with IPSG enabled on the incoming interface.

  - IPSG drops ARP packets at HSRP standby.

  - With DHCP snooping and IPSG enabled, if a binding entry exists for the host, traffic is forwarded to the host even without ARP.

- Beginning with Cisco NX-OS Release 9.3(5), IP Source Guard is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

- IP Source Guard does not require TCAM carving on the Cisco Nexus 9300-X Cloud Scale Switches.

# Default Settings for IP Source Guard

This table lists the default settings for IP Source Guard parameters.

*Table 40: Default IP Source Guard Parameters*

| Parameters | Default |
|---|---|
| IP Source Guard | Disabled on each interface |
| IP source entries | None. No static or default IP source entries exist by default. |

# Configuring IP Source Guard

## Enabling or Disabling IP Source Guard on a Layer 2 Interface

You can enable or disable IP Source Guard on a Layer 2 interface. By default, IP Source Guard is disabled on all interfaces.

**Before you begin**

Make sure that the DHCP feature and DHCP snooping are enabled.

Make sure that the ACL TCAM region size for IPSG (ipsg) is configured.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#` | Enters interface configuration mode for the specified interface. |
| **Step 3** | [no] **ip verify source dhcp-snooping-vlan**<br><br>**Example:**<br><br>`switch(config-if)# ip verify source`<br>`dhcp-snooping vlan` | Enables IP Source Guard on the interface. The **no** form of this command disables IP Source Guard on the interface. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 4** | (Optional) **show running-config dhcp**<br><br>**Example:**<br>`switch(config-if)# show running-config dhcp` | Displays the running configuration for DHCP snooping, including the IP Source Guard configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Adding or Removing a Static IP Source Entry

You can add or remove a static IP source entry on the device. By default, there are no static IP source entries.

**Procedure**

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip source binding** *ip-address mac-address* **vlan** *vlan-id* **interface** *interface-type slot/port*<br><br>**Example:**<br>`switch(config)# ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface ethernet 2/3` | Creates a static IP source entry for the current interface. The **no** form of this command removes the static IP source entry. |
| **Step 3** | (Optional) **show ip dhcp snooping binding** [**interface** *interface-type slot/port*]<br><br>**Example:**<br>`switch(config)# show ip dhcp snooping binding interface ethernet 2/3` | Displays IP-MAC address bindings for the interface specified, including static IP source entries. Static entries appear with the term in the Type column. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring IP Source Guard for Trunk Ports

When IP Source Guard is configured on a port, traffic coming on that port will be dropped unless there is a DHCP snooping entry to allow it in the TCAM. However, when IP Source Guard is configured on trunk ports and you do not want traffic coming on certain VLANs to undergo this check (even if DHCP snooping is not enabled on them), you can specify a list of VLANs to exclude.

### Before you begin

Make sure that the DHCP feature and DHCP snooping are enabled.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **ip dhcp snooping ipsg-excluded vlan** *vlan-list*<br>**Example:**<br>`switch(config)# ip dhcp snooping`<br>`ipsg-excluded vlan 1001-1256,3097` | Specifies the list of VLANs to exclude from the DHCP snooping check for IP Source Guard on trunk ports. |
| **Step 3** | (Optional) **show ip ver source** [**ethernet** *slot/port* \| **port-channel** *channel-number*]<br>**Example:**<br>`switch(config)# show ip ver source` | Displays which VLANs are excluded. |
| **Step 4** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Displaying IP Source Guard Bindings

Use the **show ip ver source** [**ethernet** *slot/port* \| **port-channel** *channel-number*] command to display the IP-MAC address bindings.

# Clearing IP Source Guard Statistics

To clear IP Source Guard statistics, use the commands in this table.

| Command | Purpose |
|---|---|
| **clear access-list ipsg stats** [**instance** *number* \| **module** *number*] | Clears IP Source Guard statistics. |

# Configuration Example for IP Source Guard

This example shows how to create a static IP source entry and enable IP Source Guard on an interface:

```
ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface ethernet 2/3
interface ethernet 2/3
  no shutdown
  ip verify source dhcp-snooping-vlan
  show ip ver source

  IP source guard excluded vlans:
  -------------------------------------------------------
  None


  ----------------------------------
  IP source guard is enabled on the following interfaces:
  -------------------------------------------------------
        ethernet2/3
```

# Additional References

## Related Documents

| Related Topic | Document Title |
|---|---|
| ACL TCAM regions | Configuring IP ACLs, on page 289 |
| DHCP and DHCP snooping | Configuring DHCP, on page 421 |

**CHAPTER 20**

# Configuring Password Encryption

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

This chapter includes the following sections:

# About AES Password Encryption and Primary Encryption Keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as Type-6 encryption. To start using Type-6 encryption, you must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in Type-6 encrypted format, unless you disable Type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to Type-6 encrypted passwords.

**Related Topics**

Configuring a Primary Key and Enabling the AES Password Encryption Feature, on page 517
Configuring Global RADIUS Keys, on page 61
Configuring a Key for a Specific RADIUS Server, on page 62
Configuring Global TACACS+ Keys, on page 91
Configuring a Key for a Specific TACACS+ Server, on page 92
Configuring a Primary Key and Enabling the AES Password Encryption Feature, on page 517

# Guidelines and Limitations for Password Encryption

Password encryption has the following configuration guidelines and limitations:

- Only users with administrator privilege (network-admin) can configure the AES password encryption feature, associated encryption and decryption commands, and primary keys.

- Beginning with Cisco NX-OS Release 10.3(3)F, RPM keychain infra supports AES password encryption for RPM legacy keychains on Cisco Nexus 9000 Series platform switches.

- Configurations containing Type-6 encrypted passwords are not rollback-compliant.

- You can enable the AES password encryption feature without a primary key, however the encryption starts only when a primary key is present in the system.

- For TACACS+ and RPM legacy keychain, after you enable the AES password encryption feature and configure a primary key, you must run the **encryption re-encrypt obfuscated** command to convert the passwords to Type-6 encrypted passwords.

- Deleting the primary key stops Type-6 encryption and causes all existing Type-6 encrypted passwords to become unusable, unless the same primary key is reconfigured.

- To move the device configuration to another device, either decrypt the configuration before porting it to the other device or configure the same primary key on the device to which the configuration will be applied.

- Type-6 encryption is supported only for MACsec and RPM legacy keychain. It is not supported for cloudsec keys.

- Starting from Cisco NX-OS Release 9.3(6), converting Type-6 encrypted passwords back to original state is not supported on MACsec keychain.

- Starting from Cisco NX-OS Release 10.3(3)F, converting Type-6 encrypted passwords back to original state is not supported for RPM legacy keychain.

- Type-6 encryption can be configured only when the AES password encryption feature is enabled and the primary key is configured.

- When the primary key is configured and the AES password encryption feature is enabled on a switch, each MACsec key string configurations under the keychain infra are automatically encrypted with the Type-6 encryption.

- Primary key configuration is local to the switch. If you take the Type-6 configured running data from one switch and apply it on another switch where a different primary key is configured, then decryption on the new switch fails.

- If you erase the startup configuration and use the configuration replace feature after a Type-6 encryption, the configuration replace fails because the primary key is not stored in PSS. Therefore, there is configuration loss for MACsec Type-6 encrypted key string.

- When you configure the Type-6 keys, you cannot modify the existing Type-6 encrypted key strings to Type-7 encrypted key string without applying the decrypt command provided by SKSD.

- If you downgrade the system by cold reboot with an old image where the Type-6 encryption is not supported, you must take out the configuration before you proceed with the cold reboot. Failing to do so leads to loss in configuration.

- After you downgrade the system, the Type-6 configuration is lost.

- If you downgrade the system by ISSD, capability conf check is invoked and it notifies you to remove the configuration before proceeding with the downgrade. You can use the **encryption decrypt** command to convert the Type-6 encrypted keys to Type-7 encryption keys, and then proceed with the downgrade.

- During an ISSU upgrade, if you migrate from an older image which includes the Type-7 encrypted keys to a new image that supports Type-6 encryption, the rpm does not convert the existing keys to Type-6

encrypted keys until re-encryption is enforced. To enforce a re-encryption, use the **encryption re-encrypt obfuscated** command.

- After ISSU upgrade from an older image which includes the Type-7 encrypted keys to a new image that supports Type-6 encryption, if configuration replace is done using the configuration file saved in older image or configuration file saved after upgrade without re-encrypting the password to Type-6 (using **encryption re-encrypt obfuscated** command), the configuration replace will fail.

- If you change the primary key after a Type-6 encryption, the decrypt command fails on the existing Type-6 encrypted key-string. You must delete the existing Type-6 key string and configure a new key string.

- For RPM legacy keychains, Type-6 key-strings can be configured without AES password encryption feature enabled and primary key configured, however these Type-6 key-strings are unusable until AES password encryption feature is enabled and the primary key with which the Type-6 key-strings were generated is configured.

- Starting from Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode.

# Default Settings for Password Encryption

This table lists the default settings for password encryption parameters.

*Table 41: Default Password Encryption Parameter Settings*

| Parameters | Default |
|---|---|
| AES password encryption feature | Disabled |
| Primary key | Not configured |

# Configuring Password Encryption

This section describes the tasks for configuring password encryption on Cisco NX-OS devices.

# Configuring a Primary Key and Enabling the AES Password Encryption Feature

You can configure a primary key for Type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption is supported for RPM legacy keychain.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | [**no**] **key config-key ascii**[ <new_key> old <old_master_key>] | Configures a primary key (`Master Key`) to be used with the AES password encryption feature. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br><br>`switch# key config-key ascii`<br>`New Master Key:`<br>`Retype Master Key:` | The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.<br><br>If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.<br><br>**Note** Starting with Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode. |
| Step 2 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 3 | [**no**] **feature password encryption aes tam**<br><br>**Example:**<br><br>`switch(config)# feature password`<br>`encryption aes tam` | Enables or disables the AES password encryption feature. |
| Step 4 | **encryption re-encrypt obfuscated**<br><br>**Example:**<br><br>`switch(config)# encryption re-encrypt`<br>`obfuscated` | Converts existing plain or weakly encrypted passwords to Type-6 encrypted passwords. |
| Step 5 | (Optional) **show encryption service stat**<br><br>**Example:**<br><br>`switch(config)# show encryption service`<br>`stat` | Displays the configuration status of the AES password encryption feature and the primary key. |
| Step 6 | **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration.<br><br>**Note** This command is necessary to synchronize the primary key in the running configuration and the startup configuration. |

**Related Topics**

# Converting Existing Passwords to Type-6 Encrypted Passwords

You can convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords.

### Before you begin

Ensure that you have enabled the AES password encryption feature and configured a primary key.

### Procedure

|        | Command or Action | Purpose |
|--------|-------------------|---------|
| **Step 1** | **encryption re-encrypt obfuscated**<br><br>**Example:**<br><br>`switch# encryption re-encrypt obfuscated` | Converts existing plain or weakly encrypted passwords to Type-6 encrypted passwords. |

# Converting Type-6 Encrypted Passwords Back to Their Original States

You can convert Type-6 encrypted passwords back to their original states. This functionality is not supported for macsec keychain.

### Before you begin

Ensure that you have configured a primary key.

### Procedure

|        | Command or Action | Purpose |
|--------|-------------------|---------|
| **Step 1** | **encryption decrypt type6**<br><br>**Example:**<br><br>`switch# encryption decrypt type6`<br>`Please enter current Master Key:` | Converts Type-6 encrypted passwords back to their original states. |

# Enabling Type-6 Encryption on MACsec Keys

The type-6 encryption feature, also known as the Advanced Encryption Standard (AES) password encryption feature allows you to securely store MACsec keys in a type-6 encrypted format.

Beginning with Cisco NX-OS Release 9.3(5), you can store MACsec keys in a type-6 encrypted format on all Cisco Nexus 9000 Series switches which support the MACsec feature.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **key config-key ascii**<br><br>**Example:**<br><br>`switch(config)# key config-key ascii`<br>`switch(config)# New Master Key:`<br>`Switch(config)# Retype Master Key:` | Configures the primary key (`Master Key`). |
| **Step 3** | [**no**] **feature password encryption aes**<br><br>**Example:**<br><br>`switch(config)# feature password`<br>`encryption aes` | Enables or disables the AES password encryption feature. |
| **Step 4** | **key chain** *name* **macsec**<br><br>**Example:**<br><br>`switch(config)# key chain 1 macsec`<br>`switch(config-macseckeychain)#` | Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode. |
| **Step 5** | **key** *key-id*<br><br>**Example:**<br><br>`switch(config-macseckeychain)# key 1000`<br>`switch(config-macseckeychain-macseckey)#` | Creates a MACsec key and enters MACsec key configuration mode. The range is 1–32 octets, and the maximum size is 32 or 64 bits. AES_128 is used for 32 bit, while AES_256 is used for 64 bits. |
| **Step 6** | **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** \| **AES_256_CMAC**}<br><br>**Example:**<br><br>`switch(config-macseckeychain-macseckey)#`<br>`key-octet-string`<br>`abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789`<br>`cryptographic-algorithm AES_256_CMAC` | Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command.<br><br>The key octet string includes the following:<br><br>• 0 Encryption Type - No encryption (default)<br><br>• 6 Encryption Type - Proprietary (Type-6 encrypted)<br><br>• 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters |

# Deleting Type-6 Encrypted Passwords

You can delete all Type-6 encrypted passwords from the Cisco NX-OS device.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **encryption delete type6**<br><br>**Example:**<br><br>`switch# encryption delete type6` | Deletes all Type-6 encrypted passwords. |

# Verifying the Password Encryption Configuration

To display password encryption configuration information, perform the following task:

| **Command** | **Purpose** |
|---|---|
| **show encryption service stat** | Displays the configuration status of the AES password encryption feature and the primary key. |

# Configuration Examples for Password Encryption

The following example shows how to create a primary key, enable the AES password encryption feature, and configure a Type-6 encrypted password for a TACACS+ application:

```
key config-key ascii
  New Master Key:
  Retype Master Key:
configure terminal
feature password encryption aes tam
show encryption service stat
  Encryption service is enabled.
  Master Encryption Key is configured.
  Type-6 encryption is being used.
feature tacacs+
tacacs-server key Cisco123
show running-config tacacs+
  feature tacacs+
  logging level tacacs 5
  tacacs-server key 6
"JDYkqyIFWeBvzpljSfWmRZrmRSRE8syxKlOSjP9RCCkFinZbJI3GD5c6rckJR/Qju2PKLmOewbheAA=="
```

# Configuring Keychain Management

This chapter describes how to configure keychain management on a Cisco NX-OS device.

This chapter includes the following sections:

## About Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication. For more information, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

### Lifetime of a Key

To maintain stable communications, each device that uses a protocol that is secured by key-based authentication must be able to store and use more than one key for a feature at the same time. Based on the send and accept lifetimes of a key, keychain management provides a secure mechanism to handle key rollover. The device uses the lifetimes of keys to determine which keys in a keychain are active.

Each key in a keychain has two lifetimes, as follows:

**Accept lifetime**

The time interval within which the device accepts the key during a key exchange with another device.

**Send lifetime**

The time interval within which the device sends the key during a key exchange with another device.

You define the send and accept lifetimes of a key using the following parameters:

**Start-time**

The absolute time that the lifetime begins.

**End-time**

The end time can be defined in one of the following ways:

- The absolute time that the lifetime ends

- The number of seconds after the start time that the lifetime ends

- Infinite lifetime (no end-time)

During a key send lifetime, the device sends routing update packets with the key. The device does not accept communication from other devices when the key sent is not within the accept lifetime of the key on the device.

We recommend that you configure key lifetimes that overlap within every keychain. This practice avoids failure of neighbor authentication due to the absence of active keys.

# Prerequisites for Keychain Management

Keychain management has no prerequisites.

# Guidelines and Limitations for Keychain Management

Keychain management has the following configuration guidelines and limitations:

- Changing the system clock impacts when the keys are active.

- A keychain's configuration type must match the type it has been linked to within the client protocol. If an attempt is made to mismatch these types, a syslog message is generated to notify the user.

  For example: It is not supported if a keychain named **keychain_abc** is configured as a Macsec keychain but is associated as a Classic keychain with OSPF. Similarly, the case where the keychain is first associated with the client (a process known as forward-referencing) and then configured as a different keychain type, is also not supported.

- It is highly recommended for user to specify the passwordtype and password when programmatically(restconf/Netconf and so on) configuring a neighbor/template's password. When either one of the property is missing in the programmatic call, BGP will use already available(or default) value of the missing property to configure the neighbor/template's password.

  If the user has to configure with a property missing then the user has to follow the same sequence of steps in both peer routers.

# Default Settings for Keychain Management

This table lists the default settings for Cisco NX-OS keychain management parameters.

**Table 42: Default Keychain Management Parameters**

| Parameters | Default |
|---|---|
| Key chains | No keychain exists by default. |
| Keys | No keys are created by default when you create a new keychain. |
| Accept lifetime | Always valid. |
| Send lifetime | Always valid. |
| Key-string entry encryption | Unencrypted. |

# Configuring Keychain Management

## Creating a Keychain

You can create a keychain on the device. A new keychain contains no keys.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **key chain** *name*<br><br>**Example:**<br><br>`switch(config)# key chain bgp-keys`<br>`switch(config-keychain)#` | Creates the keychain and enters keychain configuration mode. |
| **Step 3** | (Optional) **show key chain** *name*<br><br>**Example:**<br><br>`switch(config-keychain)# show key chain`<br>` bgp-keys` | Displays the keychain configuration. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:** | Copies the running configuration to the startup configuration. |

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

# Removing a Keychain

You can remove a keychain on the device.

> **Note**    Removing a keychain removes any keys within the keychain.

### Before you begin

If you are removing a keychain, ensure that no feature uses it. If a feature is configured to use a keychain that you remove, that feature is likely to fail to communicate with other devices.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 2 | **no key chain** *name*<br><br>**Example:**<br>switch(config)# no key chain bgp-keys | Removes the keychain and any keys that the keychain contains. |
| Step 3 | (Optional) **show key chain** *name*<br><br>**Example:**<br>switch(config-keychain)# show key chain bgp-keys | Confirms that the keychain no longer exists in running configuration. |
| Step 4 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config-keychain)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Configuring a Primary Key and Enabling the AES Password Encryption Feature

You can configure a primary key for Type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption is supported for RPM legacy keychain.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | [**no**] **key config-key ascii**[ <new_key> old <old_master_key>]<br><br>**Example:**<br>`switch# key config-key ascii`<br>`New Master Key:`<br>`Retype Master Key:` | Configures a primary key (Master Key) to be used with the AES password encryption feature. The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.<br><br>If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.<br><br>**Note** Starting with Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 3** | [**no**] **feature password encryption aes tam**<br><br>**Example:**<br>`switch(config)# feature password`<br>`encryption aes tam` | Enables or disables the AES password encryption feature. |
| **Step 4** | **encryption re-encrypt obfuscated**<br><br>**Example:**<br>`switch(config)# encryption re-encrypt`<br>`obfuscated` | Converts existing plain or weakly encrypted passwords to Type-6 encrypted passwords. |
| **Step 5** | (Optional) **show encryption service stat**<br><br>**Example:**<br>`switch(config)# show encryption service`<br>`stat` | Displays the configuration status of the AES password encryption feature and the primary key. |
| **Step 6** | **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration.<br><br>**Note** This command is necessary to synchronize the primary key in the running configuration and the startup configuration. |

**Related Topics**

# Configuring Text for a Key

You can configure the text for a key. The text is the shared secret. The device stores the text in a secure format.

For MACsec and RPM legacy keychain, the text is encrypted and stored in Type-6 format if AES password encryption feature is enabled and primary key configured otherwise it will be stored in Type-7 encrypted format.

By default, accept and send lifetimes for a key are infinite, which means that the key is always valid. After you configure the text for a key, configure the accept and send lifetimes for the key.

### Before you begin

Determine the text for the key. You can enter the text as unencrypted text or in the encrypted form that Cisco NX-OS uses to display key text when you use the **show key chain** command. Using the encrypted form is particularly helpful if you are creating key text to match a key as shown in the **show key chain** command output from another device.

### Procedure

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **key chain** *name*<br><br>**Example:**<br><br>`switch(config)# key chain bgp-keys`<br>`switch(config-keychain)#` | Enters keychain configuration mode for the keychain that you specified. |
| **Step 3** | **key** *key-ID*<br><br>**Example:**<br><br>`switch(config-keychain)# key 13`<br>`switch(config-keychain-key)#` | Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535. |
| **Step 4** | **key-string** [*encryption-type*] *text-string*<br><br>**Example:**<br><br>`switch(config-keychain-key)# key-string`<br>` 0 AS3cureStr1ng` | Configures the text string for the key. The *text-string* argument is alphanumeric, case-sensitive, and supports special characters.<br><br>The *encryption-type* argument can be one of the following values:<br><br>    • 0—The *text-string* argument that you enter is unencrypted text. This is the default. |

| | Command or Action | Purpose |
|---|---|---|
| | | • 6—Beginning with Cisco NX-OS Release 10.3(3)F, the Cisco proprietary (Type-6 encrypted) method is supported on Cisco Nexus 9000 Series platform switches. |
| | | • 7—The *text-string* argument that you enter is encrypted. The encryption method is a Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Cisco NX-OS device. |

The **key-string** command has limitations on using the following special characters in the *text-string*:

| Special Character | Des |
|---|---|
| \| | Ver |
| > | Gre |
| \ | Bac |
| ( | Lef |
| ' | Apc |
| " | Quc |
| ? | Quc |

For more information on the special characters usage in commands, see Understanding the Command-Line Interface section.

| | Command or Action | Purpose |
|---|---|---|
| Step 5 | (Optional) **show key chain** *name* [**mode decrypt**]<br><br>**Example:**<br>`switch(config-keychain-key)# show key chain bgp-keys` | Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-keychain-key)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Configuring a Primary Key and Enabling the AES Password Encryption Feature, on page 517

# Configuring Accept and Send Lifetimes for a Key

You can configure the accept lifetime and send lifetime for a key. By default, accept and send lifetimes for a key are infinite, which means that the key is always valid.

> ✎
>
> **Note** We recommend that you configure the keys in a keychain to have overlapping lifetimes. This practice prevents loss of key-secured communication due to moments where no key is active.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **key chain** *name*<br><br>**Example:**<br>`switch(config)# key chain bgp-keys`<br>`switch(config-keychain)#` | Enters keychain configuration mode for the keychain that you specified. |
| **Step 3** | **key** *key-ID*<br><br>**Example:**<br>`switch(config-keychain)# key 13`<br>`switch(config-keychain-key)#` | Enters key configuration mode for the key that you specified. |
| **Step 4** | **accept-lifetime** [**local**] *start-time* [**duration** *duration-value* \| **infinite** \| *end-time*]<br><br>**Example:**<br>`switch(config-keychain-key)#`<br>`accept-lifetime 00:00:00 Jun 13 2013`<br>`23:59:59 Sep 12 2013` | Configures an accept lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.<br><br>The *start-time* argument is the time of day and date that the key becomes active.<br><br>Specify the end of the lifetime with one of the following options:<br><br>• **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).<br><br>• **infinite**—The accept lifetime of the key never expires.<br><br>• *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive. |

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 5** | **send-lifetime** [**local**] *start-time* [**duration** *duration-value* \| **infinite** \| *end-time*]<br><br>**Example:**<br><br>`switch(config-keychain-key)# send-lifetime 00:00:00 Jun 13 2013 23:59:59 Aug 12 2013` | Configures a send lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.<br><br>The *start-time* argument is the time of day and date that the key becomes active.<br><br>You can specify the end of the send lifetime with one of the following options:<br><br>• **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).<br><br>• **infinite**—The send lifetime of the key never expires.<br><br>• *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive. |
| **Step 6** | (Optional) **show key chain** *name* [**mode decrypt**]<br><br>**Example:**<br><br>`switch(config-keychain-key)# show key chain bgp-keys` | Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-keychain-key)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

Configuring a Primary Key and Enabling the AES Password Encryption Feature, on page 517

# Configuring a Key for OSPFv2 Cryptographic Authentication

You can configure message digest 5 (MD5) or hash-based message authentication code secure hash algorithm (HMAC-SHA) authentication for OSPFv2.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| Step 2 | **key chain** *name*<br><br>**Example:**<br>```<br>switch(config)# key chain bgp-keys<br>switch(config-keychain)#<br>``` | Enters keychain configuration mode for the keychain that you specified. |
| Step 3 | **key** *key-ID*<br><br>**Example:**<br>```<br>switch(config-keychain)# key 13<br>switch(config-keychain-key)#<br>``` | Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.<br><br>**Note**    For OSPFv2, the key identifier in the key key-id command supports values from 0 to 255 only. |
| Step 4 | **[no] cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | HMAC-SHA-384 | HMAC-SHA-512 | MD5}**<br><br>**Example:**<br>```<br>switch(config-keychain-key)#<br>cryptographic-algorithm HMAC-SHA-1<br>``` | Configures the OSPFv2 cryptographic algorithm to be used for the specified key. You can configure only one cryptographic algorithm per key. |
| Step 5 | (Optional) **show key chain** *name*<br><br>**Example:**<br>```<br>switch(config-keychain-key)# show key<br>chain bgp-keys<br>``` | Shows the keychain configuration. |
| Step 6 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>```<br>switch(config-keychain-key)# copy<br>running-config startup-config<br>``` | Copies the running configuration to the startup configuration. |

# Determining Active Key Lifetimes

To determine which keys within a key chain have active accept or send lifetimes, use the command in this table.

| Command | Purpose |
|---|---|
| **show key chain** | Displays the key chains configured on the device. |

# Verifying the Keychain Management Configuration

To display keychain management configuration information, perform the following task:

| Command | Purpose |
|---|---|
| **show key chain** *name* | Displays the keychains configured on the device. |

# Configuration Example for Keychain Management

This example shows how to configure a keychain named "ospf-keys". Each key text string is encrypted. The keys are configured to use MD5 as their cryptographic algorithm. Each key has longer accept lifetimes than send lifetimes, resulting in overlap between a pair of keys. In this example, there is configured overlap between key 1 and key 2, as well as key 2 and key 3. This prevents a period of time in which there are no active keys, helping to avoid a disruption in communication of the underlying protocol:

```
key chain ospf-keys
  key 1
    key-string 7 070c285f4d0658544541
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
    cryptographic-algorithm MD5
  key 2
    key-string 7 070c285f4d0658574446
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    cryptographic-algorithm MD5
  key 3
    key-string 7 070c285fad0622474941
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    cryptographic-algorithm MD5
```

This example shows how to configure a keychain named "bgp-keys" with Type-6 encryption. This encryption mode is available when feature password encryption aes is enabled:

```
key chain bgp-keys
  key 1
    key-string 6
JDYkbN6ZTz3Hqrv5ZWliyxqlYiQXYc0wWpOnK7epMGoHK6qVJPeJtSYAGhQ9V+QKG4ZrcWeuunTtAA==
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
  key 2
    key-string 6
JDYkO6Di45Bu1ikPja/r8VJNoSTa4I4QMxtzzG3DQzal9G9LJA6F1WNGX8GRgn95SPuf4naoTZCtAA==
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
  key 3
    key-string 6
JDYk8DJ15ZdOQ/O7vnj2M92lRiR2x8VrL0Muj/30TNlIK5f+JMFEHoWy0Rfuy827G/H10w2it7eVAA==
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
```

# Where to Go Next

For information about routing features that use keychains, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

# Additional References for Keychain Management

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Border Gateway Protocol | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |
| OSPFv2 | *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* |

**Standards**

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

# Configuring Traffic Storm Control

This chapter describes how to configure traffic storm control on the Cisco NX-OS device.

This chapter includes the following sections:

## About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Layer 2 or Layer 3 ports by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a 3.9-millisecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

This table shows the broadcast traffic patterns on a Layer 2 or Layer 3 interface over a given interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

*Figure 19: Broadcast Suppression*



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Cisco Nexus 9000v device is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 or Layer 3 interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 3.9-millisecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 3.9-millisecond interval can affect the behavior of traffic storm control.

The following are examples of how traffic storm control operation is affected

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast traffic until the end of the interval.

- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

When the traffic exceeds the configured level, you can configure traffic storm control to perform the following optional corrective actions :

- Shut down—When ingress traffic exceeds the traffic storm control level that is configured on a port, traffic storm control puts the port into the error-disabled state. To reenable this port, you can use either the **shutdown** and **no shutdown** options on the configured interface, or the error-disable detection and recovery feature. You are recommended to use the **errdisable recovery cause storm-control** command for error-disable detection and recovery along with the **errdisable recovery interval** command for defining the recovery interval. The interval can range between 30 and 65535 seconds.

- Trap—You can configure traffic storm control to generate an SNMP trap when ingress traffic exceeds the configured traffic storm control level. The SNMP trap action is enabled by default. However, storm

control traps are not rate-limited by default. You can control the number of traps generated per minute by using the **snmp-server enable traps storm-control trap-rate** command.

By default, Cisco NX-OS takes no corrective action when traffic exceeds the configured level.

# Licensing Requirements for Traffic Storm Control

The following table shows the licensing requirements for this feature:

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

# Guidelines and Limitations for Traffic Storm Control

Traffic storm control has the following configuration guidelines and limitations:

- The storm control feature does not work if you enable storm control on an interface where sFlow is also enabled.

- Storm control PPS option was supported only on Cisco Nexus 9300-FX2 platform switches. Beginning with Cisco NX-OS Release 10.3(2)F, it is also supported on Cisco Nexus 9300-FX3, 9300-GX, and 9300-GX2 platform switches.

- For Cisco Nexus NFE2-enabled devices, you can use the storm control-cpu to control the number of ARP packets sent to the CPU.

- Storm control can be configured on physical, port-channel, and breakout interfaces.

- Specify the traffic storm control level as a percentage of the total interface bandwidth:
    - The pps range can be from 0 to 200000000.
    - The optional fraction of a level can be from 0 to 99.
    - 100 percent means no traffic storm control.
    - 0.0 percent suppresses all traffic.

- For Cisco Nexus 9500 Series switches with 9400 Series line cards, and Cisco Nexus 9300 Series switches, you can use the storm control CLI to specify bandwidth level either as a percentage of port capacity or packets-per-second.

- Beginning with Cisco Nexus Release 9.2(1), the error margin is greater than 1% when you configure the storm control packets-per-seconds as follows:
    - Traffic period < 60 s
    - Storm control pps <1000

- Beginning with Cisco Nexus Release 9.2(1), you can use the percentage of port capacity or packets-per-second for the Cisco Nexus 9336C-FX2, Cisco Nexus 93300YC-FX2, and Cisco Nexus 93240YC-FX2-Z switches.

- If you have configured an SVI for the VLAN on Cisco Nexus 9200, 9300-EX platform switches, or on the N9K-X9700-FX3 line cards, storm control broadcast does not work for ARP traffic (ARP request).

- Local link and hardware limitations prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the discards counter.

- Because of hardware limitations and the method by which packets of different sizes are counted, the traffic storm control level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

- Due to a hardware limitation, the output for the **show interface counters storm-control** command does not show ARP suppression when storm control is configured and the interface is actually suppressing ARP broadcast traffic. This limitation can lead to the configured action not being triggered but the incoming ARP broadcast traffic being correctly storm suppressed.

- Due to a hardware limitation, storm control is not supported for 400G ports beyond 70% of the port bandwidth in Cisco Nexus GX series platform switches.

- Due to a hardware limitation, the packet drop counter cannot distinguish between packet drops caused by a traffic storm and packet drops caused by other discarded input frames. This limitation can lead to the configured action being triggered even in the absence of a traffic storm.

- Due to a hardware limitation, storm suppression packet statistics are not supported on uplink ports.

- Due to a hardware limitation, storm suppression packet statistics do not include broadcast traffic on VLANs with an active switched virtual interface (SVI).

- Due to a design limitation, storm suppression packet statistics do not work if the configured level is 0.0, which is meant to suppress all incoming storm packets.

- Traffic storm control is supported on the Cisco Nexus 9300 Series switches and the Cisco Nexus 9500 Series switches with the 9700-EX/FX line card.

- Traffic storm control is not supported on Cisco N9K-M4PC-CFP2.

- Traffic storm control is not supported on FEX interfaces.

- Traffic storm control is only for ingress traffic, specifically for unknown unicast, unknown multicast, and broadcast traffic.

**Note** On Cisco Nexus 9000 Series switches, traffic storm control applies to unknown unicast traffic and not known unicast traffic

- When port channel members are error disabled due to a configured action, all individual member ports should be flapped to recover from the error disabled state.

- Cisco Nexus Release 9.2(1) the traffic storm control feature is not supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R line card and N9K-C9504-FM-R fabric module.

- Beginning with Cisco Nexus Release 9.2(4), the traffic storm control feature is supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R line card and N9K-C9504-FM-R fabric module. Traffic storm control counters do not increment when the interface is flooded with the broadcast traffic.

- Beginning with Cisco Nexus Release 9.3(2), the traffic storm control feature with only rate-limiting is supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R, N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards, and N9K-C9504-FM-R and N9K-C9508-FM-R fabric modules. Traffic storm control counters and storm-control action are not supported.

- The following guidelines and limitations apply to Cisco Nexus 9200 Series switches:

  - Traffic storm control with unknown multicast traffic is not supported.

  - Packet-based statistics are not supported for traffic storm control as the policer supports only byte-based statistics.

  - Traffic storm control is not supported for copy-to-CPU packets.

- Beginning with Cisco NX-OS Release 10.1(2), Storm Control feature is supported on the N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.

- Beginning with Cisco Nexus Release 10.1(2), for Cisco Nexus N9300-FX and N9300-FX2 series switches, you can configure a two-level threshold and logging support for Broadcast, Unknown Unicast, and Multicast (BUM) traffic, and also set trap or shutdown action for each threshold level. The existing storm control configuration is now used only for one-level threshold.

- The following guidelines and limitations apply to the two-level threshold and logging support for BUM traffic feature for Cisco Nexus 10.1(2) release:

  - The new traffic storm control feature in Cisco Nexus Release 10.1(2) supports a maximum of 62 ports (as a single slice) on Cisco Nexus N9300-FX and a total of 124 ports (as two slices) on Cisco Nexus N9300-FX2.

  - Traffic storm control supports devices that are only in one storm control mode at a time, either one-level or two-level threshold. It does not support a mix of one-level threshold and two-level threshold storm control mode across ports at a time.

  - Traffic storm control monitors traffic statistics and generates system log for each level (lower and higher) and traffic type (unknown unicast, multicast, and broadcast) from Cisco Nexus Release 10.1(2).

  - The two-level threshold traffic storm control feature requires carving of a new Ternary Content Addressable Memory (TCAM) region with a fixed size of 512, and a reload of the device.

  - Traffic storm control for two-level threshold cannot coexist with the L2 Netflow feature, that is, presence of config layer2-switched flow monitor CLI, because of TCAM resource limitation.

  - The two-level threshold feature for traffic storm control does not support non-IP MC flood traffic (packet without an IP header) and packets-per-second mode.

  - Traffic storm control is not supported on Generic Online Diagnostics (GOLD) packets and sub-interface level.

  - If you were on a prior release, have upgraded to 10.1(2), and want to use the two-level storm control feature, then make sure that you configure the switch with the new storm control commands.

- If you have configured the two-level storm control feature in version 10.1(2), and you want to downgrade to a previous version, then the new feature does not support downgrade. To downgrade, remove the configuration.

- Beginning from Cisco Nexus Release 10.2(1), Storm control does not allow to have multiple action configurations on an interface. If the previous action value is overwritten, then it considers the latest action value that is configured.

- Beginning with Cisco NX-OS Release 10.2(2)F, the storm control feature is supported on Cisco N9K-9332D-GX2B platform switches.

- Beginning with Cisco NX-OS Release 10.3(2)F, Traffic Storm Control is supported on Layer 3 interfaces, and the following guidelines and limitations are applicable:

  - Traffic storm control supports devices that are only on one-level threshold storm control mode.

  - Layer 3 packets destined for the control plane such as ARP broadcast are not suppressed by storm control and are policed by the CoPP policer. However, storm control violation actions are triggered.

  - This feature is supported only on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and Cisco Nexus 9500 Series switches with FX and GX line cards.

- Beginning with Cisco NX-OS Release 10.3(3)F, the **system storm-control priority-policy drop-l3** command is introduced to prioritize storm control drop over the CoPP policer, and the following guidelines and limitations are applicable:

  - This feature applies to Layer 3 control frames.

  - This feature is supported only on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and Cisco Nexus 9500 Series switches with FX and GX line cards.

  - This feature is applicable only for one-level threshold traffic storm control.

# Default Settings for Traffic Storm Control

This table lists the default settings for traffic storm control parameters.

*Table 43: Default Traffic Storm Control Parameters*

| Parameters | Default |
|---|---|
| Traffic storm control | Disabled |
| Threshold percentage | 100 |

# Configuring Traffic Storm Control for One-level Threshold

You can set the percentage of total available bandwidth that the controlled traffic can use for one-level threshold.

✎

| Note | • Traffic storm control uses a 3.9-millisecond interval that can affect the behavior of traffic storm control. |
| | • You must carve the n9k-arp-acl TCAM region before setting storm-control-cpu rate on port-channel. For information on configuring the TCAM region size, see the *Configuring ACL TCAM Region Sizes* section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. |

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface** {**ethernet** *slot*/*port* \| **port-channel** *number*}<br><br>**Example:**<br>`switch# interface ethernet 1/1`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | [**no**] **storm-control** {**broadcast** \| **multicast** \| **unicast**} **level** { *<level-value %>* \| **pps** < *pps-value* > }<br><br>**Example:**<br>`switch(config-if)# storm-control unicast level 40`<br><br>**Example:**<br>`switch(config-if)# storm-control broadcast level pps 8000` | Configures traffic storm control for traffic on the interface. You can also configure bandwidth level as a percentage either of port capacity or packets-per-second. The default state is disabled. |
| **Step 4** | [**no**] **storm-control action trap**<br><br>**Example:**<br>`switch(config-if)# storm-control action trap` | Generates an SNMP trap (defined in CISCO-PORT-STORM-CONTROL-MIB) and a syslog message when the traffic storm control limit is reached. |
| **Step 5** | [**no**] **storm-control-cpu arp rate**<br><br>**Example:**<br>`switch(config-if)# storm-control-cpu arp rate` | Configures traffic storm control rate for arp packets entering a port channel. This rate is divided equally among the members of the port channel. |
| **Step 6** | **exit**<br><br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |

|  | Command or Action | Purpose |
|---|---|---|
| Step 7 | (Optional) **show running-config interface** {**ethernet** *slot*/*port* \| **port-channel** *number*}<br><br>**Example:**<br>`switch(config)# show running-config interface ethernet 1/1` | Displays the traffic storm control configuration. |
| Step 8 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Prioritizing Storm-control Policer Over the CoPP Policer

By default, the Layer 3 control packets such as ARP broadcast are dropped by the storm-control policer, but policed by the CoPP policer and sent to CPU. This is because, for control packets, the storm-control policer has a lower priority compared to the CoPP policer. Enabling the prioritize storm policer over the CoPP policer feature allows for increasing the priority of the storm-control policer over the CoPP policer for Layer 3 control packets. Consequently, the Layer 3 control packets such as ARP broadcast are dropped completely and not policed further by the CoPP policer, but policed by the storm-control policer.

**Note** When this feature is enabled, and if the incoming traffic is above the configured threshold value for a specific traffic type (multicast and broadcast), then the storm-control policer may not have control on the packets that get dropped. In this scenario, even the genuine Layer 3 control packets of that specific traffic type (multicast and broadcast) may get dropped.

Use the following command to prioritize the storm-control policer over the CoPP policer:

[**no**] **system storm-control priority-policy drop-l3**

Use the **no** form of this command to disable this feature.

# Configuring Traffic Storm Control for Two-level Threshold

You can set the percentage of total available bandwidth that the controlled traffic can use for two-level threshold.

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| Step 1 | **system storm control multi-threshold**<br><br>**Example:**<br>`switch# system storm control multi-threshold` | Enters global CLI. This command is required only for configuring two-level threshold. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **hardware access-list tcam region ing-storm-control 512**<br><br>**Example:**<br><br>switch# hardware access-list tcam region ing-storm-control 512 | Carves a new TCAM region with a fixed size of 512 for the two-level threshold.<br><br>After running the command, make sure that you reload the device. |
| Step 3 | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal<br>switch(config)# | Enters global configuration mode. |
| Step 4 | **interface** {**ethernet** *slot*/*port* \| **port-channel** *number*}<br><br>**Example:**<br><br>switch(config)# interface ethernet 1/1<br>switch(config-if)# | Enters interface configuration mode. |
| Step 5 | [**no**] **storm-control multiunicast** {**level1** <*level-value %*> \| **level2** <*level-value %*>}<br><br>**Example:**<br><br>switch(config-if)# storm-control multi unicast level1 5 level2 10 | Configures traffic storm control for traffic on the interface for two-level threshold.<br><br>You can also configure bandwidth level as a percentage of port capacity. The default state is disabled. |
| Step 6 | [**no**] **storm-control multi action1** {**trap** \| **shutdown**} **action2** {**trap** \| **shutdown**}<br><br>**Example:**<br><br>switch(config-if)# storm-control multi action1 trap action2 shutdown | Generates the following:<br><br>• An SNMP trap (defined in CISCO-PORT-STORM-CONTROL-MIB) to monitor the storm control.<br><br>• A syslog message when the traffic storm control limit is reached.<br><br>You can also configure the trap or shutdown action for the lower and higher level of storm control threshold. However, if you configure shutdown on lower threshold (level1) for a port, you must configure shutdown for higher threshold (level2) for that port. |
| Step 7 | **exit**<br><br>**Example:**<br><br>switch(config-if)# exit<br>switch(config)# | Exits interface configuration mode. |
| Step 8 | (Optional) **show running-config interface** {**ethernet** *slot*/*port* \| **port-channel** *number*}<br><br>**Example:**<br><br>switch(config)# show running-config interface ethernet 1/1 | Displays the traffic storm control configuration. |

|  | Command or Action | Purpose |
|---|---|---|
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Verifying Traffic Storm Control Configuration

To display traffic storm control configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show running-config interface** | Displays the traffic storm control configuration. |
| **show access-list storm-control arp-stats interface [ethernet** \| **port-channel]** *number* | Displays the storm control statistics for arp packets on the interface. |

# Monitoring Traffic Storm Control Counters

You can monitor the counters the Cisco NX-OS device maintains for traffic storm control activity for one-level and two-level thresholds.

| Command | Purpose |
|---|---|
| The following row is applicable only to one-level threshold. | |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control** | Displays the traffic storm control counters. |
| The following rows are applicable only to two-level threshold. | |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control multi-threshold** | Displays the list of the configured storm control values for all interfaces. |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control multi-threshold** | Displays the list of the configured storm control values for the interface. |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control multi-threshold unicast** | Displays the list of the unicast drops for both level1 and level2. |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control multi-threshold broadcast** | Displays the list of the broadcast drops for both level1 and level2. |
| **show interface [ethernet** *slot*/*port* \| **port-channel** *number*] **counters storm-control multi-threshold multicast** | Displays the list of the multicast drops for both level1 and level2. |

# Configuration Examples for Traffic Storm Control

The following example shows how to configure traffic storm control for one-level threshold:

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# storm-control broadcast level 40
switch(config-if)# storm-control multicast level 40
switch(config-if)# storm-control unicast level 40
switch(config)# storm-control-cpu arp rate 150
```

The following example shows how to configure traffic storm control for two-level threshold:

```
switch# system storm control multi-threshold
switch# hardware access-list tcam region ing-storm-control 512
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# storm-control multi broadcast level1 5 level2 10
switch(config-if)# storm-control multi multicast level1 5 level2 10
switch(config-if)# storm-control multi unicast level1 5 level2 10
switch(config-if)# storm-control multi action1 trap action2 shutdown
```

The following example checks the programmed configured rate and the statistics of dropped ARP packets:

```
switch(config)# sh access-list storm-control-cpu arp-stats
interface port-channel 132
slot  1
=======
--------------------------------------------------------------------------
          ARP Policer Entry Statistics
--------------------------------------------------------------------------
Interface port-channel132:
----------
Member Interface   Entry-ID  Rate    RedPacket Count    GreenPacket Count
--------------------------------------------------------------------------------
Ethernet1/35       3976      50         0                  0
--------------------------------------------------------------------------------

slot  7
=======
--------------------------------------------------------------------------
          ARP Policer Entry Statistics
--------------------------------------------------------------------------
Interface port-channel132:
----------
Member Interface   Entry-ID  Rate    RedPacket Count    GreenPacket Count
--------------------------------------------------------------------------------
```

# System Log Examples for Traffic Storm Control

The following example shows the system log for traffic storm control with one-level threshold:

- `%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured threshold , action - Trap`

The following example shows the system log for traffic storm control with two-level threshold:

- `%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured Broadcast threshold level1[10%], action - Trap`

- `%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured Broadcast threshold level1[15%], action – Shutdown`

| **Note** | The system log message includes the specific traffic type that exceeded the threshold and the level at which the traffic type reached the storm control action on an interface. |
|---|---|

# Additional References for Traffic Storm Control

This section includes additional information related to implementing traffic storm control.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |

# Configuring Unicast RPF

This chapter describes how to configure unicast reverse path forwarding (uRPF) on Cisco NX-OS devices.

This chapter includes the following sections:

## About Unicast RPF

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

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

**Note** Unicast RPF is an ingress function and is applied only on the ingress interface of a switch at the upstream end of a connection.

Unicast RPF verifies that any packet received at a switch interface arrives on the best return path (return route) to the source of the packet by doing a reverse lookup in the FIB. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same

interface from which the packet was received, the source address might have been modified by the attacker. If unicast RPF does not find a reverse path for the packet, the packet is dropped.

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

# Unicast RPF Process

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB that matches the route to the receiving interface. Static routes, network statements, and dynamic routing add routes to the FIB.

- IP source addresses at the receiving interface must match the routing entry for the interface.

- Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

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

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

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

1. Checks the input ACLs on the inbound interface.

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

3. Conducts a FIB lookup for packet forwarding.

4. Checks the output ACLs on the outbound interface.

5. Forwards the packet.

# Guidelines and Limitations for Unicast RPF

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

- uRPF is supported for the following platforms:

  - Cisco Nexus 9500 Series switches with N9K-X9636C-R and N9K-X9636Q-R line cards

- Cisco Nexus 9500 Series switches with N9K-X9636C-RX line cards

- Cisco Nexus 9300 platform switches (excluding the 9300-FXP switches)

- Beginning with Cisco NX-OS Release 10.1(2), uRPF is supported on:

  - Cisco Nexus 9300-GX/GX2 series switches and Cisco Nexus 9500 series switches with FX linecards (for IPv4 and IPv6)

  - Cisco Nexus 9500 series switches with EX linecards (for IPv4 only)

  - ToR and EoR switches that support vPC

- Beginning with Cisco NX-OS Release 9.2(1), uRPF is supported on:

  - Cisco Nexus 9300-EX Series switches (for IPv4 only)

  - Cisco Nexus 9300-FX/FX2 Series switches (for IPv4 and IPv6)

- Beginning with Cisco NX-OS Release 9.3(5), uRPF is supported on Cisco Nexus 9300-FX3 platform switches (for IPv4 and IPv6).

- Beginning with Cisco Nexus Release 9.3(1), uRPF is supported on Cisco Nexus 9500 Series switches with the family of modular EX/FX line cards (see Cisco Nexus 9500 Cloud-Scale Line Cards and Fabric Modules Data Sheet ).

**Note**
uRPF on the modular EX/FX line cards is supported only in DUAL STACK MCAST routing mode. Specify the following configuration before enabling uRPF: `system routing template-dual-stack-mcast`. Refer to the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* on how to configure DUAL STACK MCAST routing mode.

From Cisco NX-OS Release 10.1(2), uRPF on the modular EX/FX line cards is supported in default routing mode, too.

- You must apply uRPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.

- The further downstream that you apply uRPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying uRPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying uRPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying uRPF across many sites does add to the administration cost of operating the network.

- The more entities that deploy uRPF across Internet, intranet, and extranet resources mean the better the chances of mitigating large-scale network disruptions throughout the Internet community and of tracing the source of an attack.

- uRPF won't inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. Configure uRPF at a home gateway so that uRPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

- You can use uRPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.

- Don't use uRPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure uRPF only where there is natural or configured symmetry.

- uRPF allows packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.

- When uRPF is enabled, the amount of static routes to null0 the switch can install is limited to the value of "Max V4 Ucast DA TCAM table entries" in "show hardware internal forwarding table utilization".

- Beginning with Cisco NX-OS Release 9.2(1), for N9K-X9636C-R and N9K-X96136YC-R switches, you can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. However, this enables Unicast RPF for both IPv4 and IPv6.

- The following guidelines and limitations apply only to Cisco Nexus 9500 Series switches with a N9K-X9636C-R, N9K-X9636C-RX, or N9K-X9636Q-R line card:

  - For strict uRPF to work, enable it on the ingress interface and the interface where the source IP address is learned.

  - The switch hardware does not implement strict uRPF per the configured routing interface.

  - Strict uRPF is implemented per learned route on strict uRPF-enabled interfaces.

  - If a route is resolved as ECMP, strict uRPF falls back to loose mode.

  - Because of the hardware limitation on the trap resolution, uRPF might not be applied on supervisor-bound packets via inband.

  - For IP traffic, enable IPv4 and IPv6 configurations simultaneously.

  - Due to hardware limitations, the N9K-X9636C-R, N9K-X9636C-RX, and N9K-X9636Q-R line cards support only the following combinations:

| uRPF Configuration | | Applied Traffic Check on Source IP Address | | | |
|---|---|---|---|---|---|
| IPv4 | IPv6 | IP Unipath | IP ECMP | MPLS Encap/VPN/ECMP | Unipath MPLS VPN for N9K-X9636C-RX Line Card |
| Disable | Disable | Allow | Allow | Allow | Allow |
| Loose | Loose | uRPF loose | uRPF loose | uRPF loose | uRPF strict |
| Strict | Strict | uRPF strict | uRPF loose | uRPF loose | uRPF strict |

- Strict uRPF discards the ICMPv6 NA packets even if the destined interface receives them for the following Cisco NX-OS devices:

  - Line cards: N9K-X9564PX, N9K-X9564TX, N9K-X9536PQ, X9408PC-CFP2, X9464TX, X9464TX2

- Uplink modules: N9K-M12PQ

- Switches: 93128TX, 9396PX, 9396TX, 9372PX, 9372PX-E, 3164Q, 31128PQ

- Strict uRPF blocks the ICMP traffic destined to the interface through VxLAN for the following platforms:
  - Cisco Nexus 9200 platform switches
  - Cisco Nexus 9300-EX/FX/GX platform switches
  - Nexus 9500 switches with N9K-X9700-EX and N9K-X9700-FX line cards

- If Strict uRPF is configured, append the following commands for urpf strict mode to work for unresolved host behind a subnet:
  - **no system multicast dcs-check**
  - **hardware profile multicast max-limit lpm-entries 0**

- Beginning with Cisco NX-OS Release 10.5(2)F, uRPF is supported on Cisco Nexus 9800 Series switches and N9K-C9336C-K1 switches with the following limitations:
  - If a route is resolved as ECMP, strict uRPF falls back to loose mode.
  - In uRPF loose mode, the **allow-default** keyword is not supported.
  - There is no separate uRPF control for IPv4 and IPv6. Configuring the uRPF feature through the command will simultaneously enable it for both IPv4 and IPv6 packets.
  - uRPF configuration is not supported on tunnel interfaces.

# Default Settings for Unicast RPF

This table lists the default settings for unicast RPF parameters.

*Table 44: Default Unicast RPF Parameter Settings*

| Parameters | Default |
|---|---|
| Unicast RPF | Disabled |

# Configuring Unicast RPF for Cisco Nexus 9500 Switches with -R Line Cards

You can configure unicast RPF on an ingress interface for Cisco Nexus 9500 Series switches with an -R line card.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | {**ip** | **ipv6**} **address** *ip-address/length*<br><br>**Example:**<br><br>`switch(config-if)# ip address`<br>`172.23.231.240/23` | Specifies an IPv4 or IPv6 address for the interface. |
| **Step 4** | {**ip** | **ipv6**} **verify unicast source reachable-via any**<br><br>**Example:**<br><br>`switch(config-if)# ip verify unicast`<br>`source reachable-via any` | Configures unicast RPF on the interface for both IPv4 and IPv6.<br><br>**Note** When you enable uRPF for IPv4 or IPv6 (using the **ip** or **ipv6** keywords), uRPF is enabled for both IPv4 and IPv6. |
| **Step 5** | (Optional) **show ip interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# show ip interface`<br>`ethernet 2/3` | Displays the IP information for an interface. |
| **Step 6** | (Optional) **show running-config interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# show running-config`<br>`interface ethernet 2/3` | Displays the configuration for an interface in the running configuration. |
| **Step 7** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuring Unicast RPF for Cisco Nexus 9300 Switches

You can configure one of the following Unicast RPF modes on an ingress interface for Cisco Nexus 9300 platform switches (excluding the 9300-FXP switches) running Cisco NX-OS Release 9.2(1) or a later release.

**Strict Unicast RPF mode**

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

**Loose Unicast RPF mode**

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **[no] system urpf disable**<br><br>**Example:**<br><br>`switch(config)# no system urpf disable` | Enables Unicast RPF on the switch.<br><br>**Note**    You must reload the Cisco NX-OS box to apply the Unicast RPF configuration. |
| **Step 3** | **interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 2/3`<br>`switch(config-if)#` | Specifies an Ethernet interface and enters interface configuration mode. |
| **Step 4** | {**ip** \| **ipv6**} **address** *ip-address/length*<br><br>**Example:**<br><br>`switch(config-if)# ip address`<br>`172.23.231.240/23` | Specifies an IPv4 or IPv6 address for the interface. |
| **Step 5** | {**ip** \| **ipv6**} **verify unicast source reachable-via** {**any** [**allow-default**] \| **rx**}<br><br>**Example:**<br><br>`switch(config-if)# ip verify unicast`<br>`source reachable-via any` | Configures Unicast RPF on the interface for both IPv4 and IPv6.<br><br>You can enable IPv4 and IPv6 uRPF separately for the Cisco Nexus 9300-EX Series switches (for IPv4) and on Cisco Nexus 9300-FX/FX2 Series switches. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** When you enable Unicast RPF for IPv4 or IPv6 (using the **ip** or **ipv6** keyword), Unicast RPF is enabled for both IPv4 and IPv6. |
| | | You can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. When you configure one version, all the mode changes must be done by this version and all other versions will be blocked by that interface. |
| | | • The **any** keyword specifies loose Unicast RPF. |
| | | • If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification. |
| | | **Note** The **allow-default** keyword is not applicable in the ALPM routing mode. |
| | | **Note** The source address lookup (in case of a loose Unicast RPF check) does not match the default route if you do not specify the **allow-default** keyword. |
| | | • The **rx** keyword specifies strict Unicast RPF. |
| **Step 6** | **exit**<br><br>**Example:**<br><br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |
| **Step 7** | (Optional) **show ip interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# show ip interface`<br>`ethernet 1/54 | grep -i "unicast reverse`<br>` path forwarding"`<br>`IP unicast reverse path forwarding: none` | Displays the IP information for an interface and verifies if Unicast RPF is enabled. |
| **Step 8** | (Optional) **show running-config interface ethernet** *slot*/*port*<br><br>**Example:**<br><br>`switch(config)# show running-config`<br>`interface ethernet 2/3` | Displays the configuration for an interface in the running configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Configuration Examples for Unicast RPF

The following example shows how to configure loose unicast RPF for IPv4 packets on a Cisco Nexus 9500 Series switch with an -R line card:

```
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure loose unicast RPF for IPv6 packets on a Cisco Nexus 9500 Series switch with an -R line card:

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

The following example shows how to configure loose unicast RPF for IPv4 packets on a Cisco Nexus 9300 platform switch:

```
no system urpf disable
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure loose unicast RPF for IPv6 packets on a Cisco Nexus 9300 platform switch:

```
no system urpf disable
interface Ethernet2/1
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via any
```

The following example shows how to configure strict unicast RPF for IPv4 packets on a Cisco Nexus 9300 platform switch:

```
no system urpf disable
interface Ethernet2/2
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
```

The following example shows how to configure strict unicast RPF for IPv6 packets on a Cisco Nexus 9300 platform switch:

```
no system urpf disable
interface Ethernet2/4
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via rx
```

# Verifying the Unicast RPF Configuration

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

| Command | Purpose |
|---|---|
| **show running-config interface ethernet** *slot*/*port* | Displays the interface configuration in the running configuration. |
| **show running-config ip** [**all**] | Displays the IPv4 configuration in the running configuration. |
| **show running-config ipv6** [**all**] | Displays the IPv6 configuration in the running configuration. |
| **show startup-config interface ethernet** *slot*/*port* | Displays the interface configuration in the startup configuration. |
| **show startup-config ip** | Displays the IP configuration in the startup configuration. |

# Additional References for Unicast RPF

This section includes additional information related to implementing unicast RPF.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Data Management Engine (DME)-ized commands | Cisco Nexus 3000 and 9000 Series NX-API REST SDK User Guide and API Reference |
| MPLS VPN | Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide |

# Configuring Switchport Blocking

This chapter describes how to configure switchport blocking on the Cisco NX-OS device.

This chapter includes the following sections:

## About Switchport Blocking

Occasionally, unknown multicast or unicast traffic is flooded to a switch port because a MAC address has timed out or has not been learned by the switch. Security issues could arise if unknown multicast and unicast traffic is forwarded to a switch port. You can enable switchport blocking to guarantee that no multicast or unicast traffic is flooded to the port.

## Guidelines and Limitations for Switchport Blocking

Switchport blocking has the following configuration guidelines and limitations:

- Switchport blocking applies only to egress ports while traffic storm control applies only to ingress ports.

- Switchport blocking is supported on all switched ports (including PVLAN ports) and is applied to all VLANs on which the port is forwarding.

- Switchport blocking is not supported for FEX ports.

- When you block unknown multicast or unicast traffic for a port channel, it is blocked on all ports in the port-channel group.

- Switchport blocking does not offer levels of control. It prevents the flooding of all unknown egress multicast or unicast packets on the specified port.

- Switchport blocking drops control packets that originate from the CPU on Cisco Nexus 9500 Series switches. It does not drop packets on Cisco Nexus 9300 Series switches.

# Default Settings for Switchport Blocking

This table lists the default settings for switchport blocking parameters.

**Table 45: Default Switchport Blocking Parameters**

| Parameters | Default |
|---|---|
| Switchport blocking | Disabled |

# Configuring Switchport Blocking

By default, the switch floods packets with unknown destination MAC addresses to all ports. To prevent the forwarding of such traffic, you can configure a port to block unknown multicast or unicast packets.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface** {**ethernet** *slot*/*port* \| **port-channel** *number*}<br><br>**Example:**<br><br>`switch# interface ethernet 1/1`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | [**no**] **switchport block** {**multicast** \| **unicast**}<br><br>**Example:**<br><br>`switch(config-if)# switchport block`<br>`unicast` | Prevents the flooding of unknown multicast or unicast packets on the specified interface.<br><br>Use the **no** form of this command to resume normal forwarding on the port. |
| **Step 4** | (Optional) **show interface** [**ethernet** *slot*/*port* \| **port-channel** *number*] **switchport**<br><br>**Example:**<br><br>`switch(config-if)# show interface`<br>`ethernet 1/1 switchport` | Displays the switchport blocking configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-if)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Verifying the Switchport Blocking Configuration

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

| Command | Purpose |
|---------|---------|
| **show interface switchport** | Displays the switchport blocking configuration for all interfaces. |
| **show interface** {**ethernet** *slot*/*port* \| **port-channel** *number*} **switchport** | Displays the switchport blocking configuration for the specified interface. |
| **show running-config interface** [**ethernet** *slot*/*port* \| **port-channel** *number*] | Displays the switchport blocking configuration in the running configuration. |

# Configuration Example for Switchport Blocking

The following example shows how to block multicast and unicast flooding on Ethernet interface 1/2 and how to verify the configuration:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# switchport block multicast
switch(config-if)# switchport block unicast
switch(config-if)# show running-config interface ethernet 1/2
!Command: show running-config interface Ethernet1/2
!Time: Wed Apr 15 16:25:48 2015

version 79.2(1)

interface Ethernet1/2
switchport
switchport block multicast
switchport block unicast
```

# Configuring Control Plane Policing

This chapter contains the following sections:

# About CoPP

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

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

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

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

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

**Control plane**

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

**Management plane**

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

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

Examples of DoS attacks include:

- Internet Control Message Protocol (ICMP) echo requests

- IP fragments

- TCP SYN flooding

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

- Reduced service quality (such as poor voice, video, or critical applications traffic)

- High route processor or switch processor CPU utilization

- Route flaps due to loss of routing protocol updates or keepalives

- Unstable Layer 2 topology

- Slow or unresponsive interactive sessions with the CLI

- Processor resource exhaustion, such as the memory and buffers

- Indiscriminate drops of incoming packets

⚠

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

# Control Plane Protection

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

## Control Plane Packet Types

Different types of packets can reach the control plane:

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

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

The following exceptions are possible from line cards only:

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

The following exceptions are possible from fabric modules only:

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

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

- match exception mtu-failure

Redirected packets
Packets that are redirected to the supervisor module.

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

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

## Classification for CoPP

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

## Egress CoPP

Beginning with Cisco NX-OS Release 10.2(3)F, egress CoPP is supported on the Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B, Nexus 9364C and Nexus 9332C cloudscale switches.

Egress CoPP can be applied on top of custom/default CoPP policy.

# Rate Controlling Mechanisms

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

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

You can configure the following parameters for policing:

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

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

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

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

# Dynamic and Static CoPP ACLs

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

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

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

- MAC-based static CoPP ACL substrings:

  - acl-mac-cdp-udld-vtp

  - acl-mac-cfsoe

  - acl-mac-dot1x

  - acl-mac-l2-tunnel

  - acl-mac-l3-isis

  - acl-mac-lacp

  - acl-mac-lldp

  - acl-mac-sdp-srp

  - acl-mac-stp

  - acl-mac-undesirable

- Protocol-based static CoPP ACL substrings:

  - acl-dhcp

- acl-dhcp-relay-response

- acl-dhcp6

- acl-dhcp6-relay-response

- acl-ptp

- Multicast-based static CoPP ACL substrings:

- acl-igmp

For more information on static CoPP ACLs, see Guidelines and Limitations for CoPP, on page 578.

# Default Policing Policies

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

- Strict—This policy is 1 rate and 2 color.

- Moderate—This policy is 1 rate and 2 color. The important class burst size is greater than the strict policy but less than the lenient policy.

- Lenient—This policy is 1 rate and 2 color. The important class burst size is greater than the moderate policy but less than the dense policy.

- Dense—This policy is 1 rate and 2 color. The policer CIR values are less than the strict policy.

- Skip—No control plane policy is applied. (Cisco does not recommend using the Skip option because it will impact the control plane of the network.)

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

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

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

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

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

**Related Topics**

Changing or Reapplying the Default CoPP Policy, on page 588

## Default Class Maps

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## Strict Default CoPP Policy

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

**Note** The CIR value for copied or custom CoPP profiles which are pre-existing before upgrade to a new image will have the same CIR values as before. The new CoPP profiles which are copied from default profiles in the new image will have a new CIR value.

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
```

```
  police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
class copp-system-p-class-important
  set cos 6
  police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
class copp-system-p-class-multicast-router
  set cos 6
  police cir 2600 kbps bc 128000 bytes conform transmit violate drop
class copp-system-p-class-management
  set cos 2
  police cir 10000 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-multicast-host
  set cos 1
  police cir 1000 kbps bc 128000 bytes conform transmit violate drop
class copp-system-p-class-l3mc-data
  set cos 1
  police cir 2400 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-normal
  set cos 1
  police cir 2200 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-ndp
  set cos 6
  police cir 1400 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-normal-dhcp
  set cos 1
  police cir 1300 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 1500 kbps bc 64000 bytes conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 3000 kbps bc 64000 bytes conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 280 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 150 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 150 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 150 kbps bc 128000 bytes conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 50 mbps bc 8192000 bytes conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 200 kbps bc 32000 bytes conform transmit violate drop
class copp-system-p-class-nat-flow
  set cos 7
  police cir 800 kbps bc 64000 bytes conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 400 kbps bc 32000 bytes conform transmit violate drop
class class-default
  set cos 0
  police cir 400 kbps bc 32000 bytes conform transmit violate drop
```

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

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

```
  set cos 1
  police cir 250 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-critical
  set cos 7
  police cir 19000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-important
  set cos 6
  police cir 3000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-multicast-router
  set cos 6
  police cir 3000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-management
  set cos 2
  police cir 3000 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-multicast-host
  set cos 1
  police cir 2000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l3mc-data
  set cos 1
  police cir 3000 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-ndp
  set cos 6
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp
  set cos 1
  police cir 300 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 400 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 6000 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 300 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 1500 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-nat-flow
  set cos 7
  police cir 100 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 50 pps bc 32 packets conform transmit violate drop
class class-default
  set cos 0
```

```
    police cir 50 pps bc 32 packets conform transmit violate drop
```

## Moderate Default CoPP Policy

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

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 96000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
```

```
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
```

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

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 96 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
```

```
    set cos 0
    police cir 15 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
```

## Lenient Default CoPP Policy

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

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
```

```
      set cos 1
      police cir 150 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-monitoring
      set cos 1
      police cir 150 kbps bc 256000 bytes conform transmit violate drop
    class copp-system-p-class-l2-unpoliced
      set cos 7
      police cir 50 mbps bc 8192000 bytes conform transmit violate drop
    class copp-system-p-class-undesirable
      set cos 0
      police cir 200 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-nat-flow
      set cos 7
      police cir 800 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-l2-default
      set cos 0
      police cir 400 kbps bc 64000 bytes conform transmit violate drop
    class class-default
      set cos 0
      police cir 400 kbps bc 64000 bytes conform transmit violate drop
```

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

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
```

```
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 300 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 1500 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-nat-flow
  set cos 7
  police cir 100 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 50 pps bc 64 packets conform transmit violate drop
class class-default
  set cos 0
  police cir 50 pps bc 64 packets conform transmit violate drop
```

## Dense Default CoPP Policy

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

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 4500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 370 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 2500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 300 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 600 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 350 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 750 kbps bc 128000 bytes conform transmit violate drop
```

```
        class copp-system-p-class-normal-dhcp-relay-response
          set cos 1
          police cir 750 kbps bc 128000 bytes conform transmit violate drop
        class copp-system-p-class-normal-igmp
          set cos 3
          police cir 1400 kbps bc 128000 bytes conform transmit violate drop
        class copp-system-p-class-redirect
          set cos 1
          police cir 200 kbps bc 32000 bytes conform transmit violate drop
        class copp-system-p-class-exception
          set cos 1
          police cir 200 kbps bc 32000 bytes conform transmit violate drop
        class copp-system-p-class-exception-diag
          set cos 1
          police cir 200 kbps bc 32000 bytes conform transmit violate drop
        class copp-system-p-class-monitoring
          set cos 1
          police cir 150 kbps bc 128000 bytes conform transmit violate drop
        class copp-system-p-class-l2-unpoliced
          set cos 7
          police cir 50 mbps bc 8192000 bytes conform transmit violate drop
        class copp-system-p-class-undesirable
          set cos 0
          police cir 100 kbps bc 32000 bytes conform transmit violate drop
        class copp-system-p-class-l2-default
          set cos 0
          police cir 200 kbps bc 32000 bytes conform transmit violate drop
        class class-default
          set cos 0
          police cir 200 kbps bc 32000 bytes conform transmit violate drop
```

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

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 2500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 1000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 1200 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
```

```
    police cir 150 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 2500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 50 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 750 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 25 pps bc 32 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 25 pps bc 32 packets conform transmit violate drop
```

## Packets Per Second Credit Limit

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

# Modular QoS Command-Line Interface

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

### Procedure

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

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

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

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

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

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

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

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

## CoPP and the Management Interface

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

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

**Related Topics**

# Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- We recommend that you use the strict default CoPP policy initially and then later modify the CoPP policies that are based on the data center and application requirements.

- First-generation Cisco Nexus 9000 Series switches (non -EX/FX/FX2), do not support source-based CoPP. This limitation does not exist for cloud scale ASIC-based Cisco Nexus switches.

- The **match-all** option is not supported in CoPP class-map and it always defaults to the **match-any** option.

- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features that are used in your specific environment and the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.

- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.

- All the traffic that you do not specify in the other class maps is put into the last class, the default class. Monitor the drops in this class and investigate if these drops are based on traffic that you do not want or the result of a feature that was not configured and you need to add.

- All broadcast traffic is sent through CoPP logic in order to determine which packets (for example, ARP and DHCP) must be redirected through an access control list (ACL) to the router processor. Broadcast

traffic that does not need to be redirected is matched against the CoPP logic, and both conforming and violated packets are counted in the hardware but not sent to the CPU. Broadcast traffic that must be sent to the CPU and broadcast traffic that does not need to be sent to the CPU must be separated into different classes.

- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.

- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the device. Filtering this traffic could prevent remote access to the Cisco NX-OS device and require a console connection.

- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (you cannot use the **service-policy output copp** command to the control plane interface).

- You can use the access control entry (ACE) hit counters in the hardware only for ACL logic. Use the software ACE hit counters and the **show access-lists** and **show policy-map type control-plane** commands to evaluate CPU traffic.

- The Cisco NX-OS device hardware performs CoPP on a per-forwarding-engine basis. CoPP does not support distributed policing. Therefore, you should choose rates so that the aggregate traffic does not overwhelm the supervisor module.

- If multiple flows map to the same class, individual flow statistics will not be available.

- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with other classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available.

- Before you downgrade from a Cisco NX-OS release that supports the CoPP feature to an earlier Cisco NX-OS release that supports the CoPP feature, you should verify compatibility using the **show incompatibility nxos bootflash:***filename* command. If an incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.

- You cannot disable CoPP. If you attempt to disable it, packets are rate limited at 50 packets per seconds.

- Skip CoPP policy option has been removed from the Cisco NX-OS initial setup utility because using it can impact the control plane of the network.

- Cisco Nexus 9200 Series switches support CoPP policer rates only in multiples of 10 kbps. If a rate is configured that is not a multiple of 10 kbps, the rate is rounded down. For example, the switch uses 50 kbps if a rate of 55 kbps is configured. (The **show policy-map type control-plane** command shows the user configured rate. See for more information.)

- For Cisco Nexus 9200 Series switches, ip icmp redirect, IPv6 icmp redirect, ip ICMP unreachable, ipv6 icmp unreachable, and mtu-failure use the same TCAM entry, and they will all be classified to the class map where the first exception is present in the policy. In the CoPP strict profile, they are classified to the class-exception class map. In a different CoPP policy, if the first exception is in a different class map (for example, class-exception-diag), the rest of the exceptions will be classified to the same class map.

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

- The following guidelines and limitations apply to static CoPP ACLs:

  - Only Cisco Nexus 9200 Series switches use static CoPP ACLs.

  - Static CoPP ACLs can be remapped to a different CoPP class.

- Access control entries (ACEs) cannot be modified or removed for static CoPP ACLs.

  - If a CoPP ACL has a static ACL substring, it maps to that type of traffic. For example, if the ACL includes the acl-mac-stp substring, STP traffic classifies to the class map for that ACL.

  - Static CoPP ACLs take priority over dynamic CoPP ACLs, regardless of their position in the CoPP policy, the order in which they are configured, and how they appear in the output of the **show policy-map type control-plane** command.

  - You must have static CoPP ACLs in the CoPP policy. Otherwise, the CoPP policy is rejected.

- Beginning with Cisco Nexus Release 9.2(2), Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches and Cisco Nexus 9500 platform switches support protocol ACL filtering. In this release, IPv6 ACL is not supported.

- Beginning with Cisco NX-OS Release 9.2(3), IPv6 ACL is supported for dynamic CoPP on the Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches, and Cisco Nexus 9500 platform switches.

- The protocol ACL filtering for egress CoPP has the following limitations:

  - Once the egress CoPP ACL is defined, you cannot add or remove an existing rule. This is applicable for all class-maps and policy-maps attached to the egress CoPP ACLs.

  - You cannot override the existing egress CoPP with a new policy. You must remove the existing egress CoPP before you add a new policy.

  - The deny action is not applicable.

  - Every entry is programmed in TCAM and uses a different TCAM space if two MAC or IP ACLs with the same entries are created and bound to either the same or a different class-map.

  - The maximum TCAM carving supported for the egress CoPP is 128 entries (24 entries are reserved and the remaining 104 entries are for egress CoPP, which are all double wide), which can be any of 52 (Ipv4, mac, Ipv6) entries.

  - Policer can be used to drop the traffic completely, with cir and burst as 0.

  - SNMP MIB is not supported.

- When a packet meets multiple exception conditions, CoPP matches the packet based on the order in which the CoPP ACLs are configured and matches it only against a single class. This is an expected CoPP behavior.

  Beginning with Cisco NX-OS Release 9.3(4), the UC FIB MISS exception is counted against the CoPP class (copp-system-p-class-exception). Therefore, if a packet has both, the TTL (accounted user class copp-system-p-class-exception-diag) and the UC FIM MISS exceptions, it is accounted against the UC FIB MISS exception. This behavior occurs because the order of the CoPP classes where the copp-system-p-class-exception class has an order higher than the copp-system-p-class-exception-diag class. For NX-OS releases earlier to NX-OS Release 9.3(4), the UC FIB MISS exception was not explicitly handled by the CoPP rules.

- CoPP processing comprises of 2 stages: In the first stage, the actual packet size is reused in each class policy, however when the packet enters the second stage, an internal header of 44 bytes is added. This causes an alteration in the conform or violation policies of all the CoPP classes. This limitation is applicable to Cisco Nexus 9300-FX, Nexus 9300-FX2, Nexus 9364C, Nexus 9332C, and 9300-GX platform switches.

- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus X9624D-R2 line cards and 9508-FM-R2 switches.

- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus 9364D-GX2A and 9332D-GX2B switches.

- Cloudscale IPv6 link-local BGP support requires carving > 512 ing-sup TCAM region (this requires a reload to take effect).

- Beginning with Cisco NX-OS Release 10.3(1)F, CoPP ACL is supported on Cisco Nexus 9808 switches.

    - Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

- Cisco Nexus 9808/9804 switches have the following limitations for SUP CoPP ACL support:

    - Policer rate is in multiples of 161 PPS at Stage-1.

    - There is no shaper in Stage 0.

    - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.

    - Fabrics/FMs are not involved in in-band path.

    - CoPP policy for Stage-1, Stage-2, and Stage-3 are in PPS.

    - CoPP Stage3 stats gets reset to zero after system switchover.

    - Only policer rate changes are supported in Custom CoPP.

- To avoid traffic loss during traffic impact, configure the CoPP class normal CIR value to 2200 kbps on Cisco Nexus 9300 GX/FX/FX2/FX3, 9504-FM-G, and 9508-FM-G switches and X9716D-GX line cards.

- Beginning with Cisco NX-OS Release 10.3(2)F, source IP based filtering in CoPP is supported on Cisco Nexus 9504 and 9508 modular chassis with R/RX line cards.

**Note**    For IPv6, source IP based filtering is supported up to 24b MSB.

- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on the Cisco Nexus 9332D-H2R switches.

- Beginning with Cisco NX-OS Release 10.4(2)F, CoPP ACL is supported on the Cisco Nexus 93400LD-H1, and 93108TC-FX3 switches.

- Beginning with Cisco Nexus Release 10.4(3)F, CoPP ACL is supported on the Cisco Nexus 9364C-H1 switches.

# Default Settings for CoPP

This table lists the default settings for CoPP parameters.

**Table 46: Default CoPP Parameters Settings**

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

# Configuring CoPP

This section describes how to configure CoPP.

# Configuring a Control Plane Class Map

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

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

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

**Before you begin**

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

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| **Step 2** | **class-map type control-plane** [**match-all** \| **match-any**] *class-map-name*<br><br>**Example:** | Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config)# class-map type control-plane ClassMapA<br>switch(config-cmap)# | **Note** You cannot use class-default, match-all, or match-any as class map names. |
| **Step 3** | (Optional) **match access-group name** *access-list-name*<br>**Example:**<br>switch(config-cmap)# match access-group name MyAccessList | Specifies matching for an IP ACL.<br>**Note** The permit and deny ACL keywords are ignored in the CoPP matching. |
| **Step 4** | (Optional) **match exception {ip \| ipv6} icmp redirect**<br>**Example:**<br>switch(config-cmap)# match exception ip icmp redirect | Specifies matching for IPv4 or IPv6 ICMP redirect exception packets. |
| **Step 5** | (Optional) **match exception {ip \| ipv6} icmp unreachable**<br>**Example:**<br>switch(config-cmap)# match exception ip icmp unreachable | Specifies matching for IPv4 or IPv6 ICMP unreachable exception packets. |
| **Step 6** | (Optional) **match exception {ip \| ipv6} option**<br>**Example:**<br>switch(config-cmap)# match exception ip option | Specifies matching for IPv4 or IPv6 option exception packets. |
| **Step 7** | **match protocol arp**<br>**Example:**<br>switch(config-cmap)# match protocol arp | Specifies matching for IP Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) packets. |
| **Step 8** | **exit**<br>**Example:**<br>switch(config-cmap)# exit<br>switch(config)# | Exits class map configuration mode. |
| **Step 9** | (Optional) **show class-map type control-plane** [*class-map-name*]<br>**Example:**<br>switch(config)# show class-map type control-plane | Displays the control plane class map configuration. |
| **Step 10** | (Optional) **copy running-config startup-config**<br>**Example:**<br>switch(config)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Configuring a Control Plane Policy Map

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

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

- 150 kilobits per second (kbps) with a burst of 32,000 bytes (for Cisco Nexus 9200 Series switches)

**Before you begin**

Ensure that you have configured a control plane class map.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>```switch# configure terminal<br>switch(config)#``` | Enters global configuration mode. |
| **Step 2** | **policy-map type control-plane** *policy-map-name*<br><br>**Example:**<br>```switch(config)# policy-map type control-plane ClassMapA<br>switch(config-pmap)#``` | Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive. |
| **Step 3** | **class** {*class-map-name* [**insert-before** *class-map-name2*] \| **class-default**}<br><br>**Example:**<br>```switch(config-pmap)# class ClassMapA<br>switch(config-pmap-c)#``` | Specifies a control plane class map name or the class default and enters control plane class configuration mode.<br><br>The class-default class map is always at the end of the class map list for a policy map. |
| **Step 4** | Enter one of the following commands:<br><br>- **police** [**cir**] {*cir-rate* [*rate-type*]}<br>- **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]<br>- **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]<br><br>**Example:**<br>```switch(config-pmap-c)# police cir 52000 bc 1000 packets```<br><br>**Example:**<br>```switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes``` | Specifies the committed information rate (CIR). The rate range is as follows:<br><br>- 01 to 268435456 pps (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches)<br>- 0 to 80000000000 bps/gbps/kbps/mbps (for Cisco Nexus 9200 Series switches)<br><br>**Note** The CIR rate range starts with 0. In previous releases, the CIR rate range starts with 1. A value of 0 drops the packet.<br><br>The committed burst (BC) range is as follows: |

| | Command or Action | Purpose |
|---|---|---|
| | | • 1 to 1073741 packets (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches) |
| | | • 1 to 512000000 bytes/kbytes/mbytes (for Cisco Nexus 9200 Series switches) |
| | | The **conform transmit** action transmits the packet. |
| | | **Note** You can specify the BC and conform action for the same CIR. |
| Step 5 | (Optional) **logging drop threshold** [*drop-count* [**level** *syslog-level*]]<br><br>**Example:**<br>`switch(config-pmap-c)# logging drop threshold 100` | Specifies the threshold value for dropped packets and generates a syslog if the drop count exceeds the configured threshold. The range for the *drop-count* argument is from 1 to 8000000000 bytes. The range for the *syslog-level* argument is from 1 to 7, and the default level is 4. |
| Step 6 | (Optional) **set cos** *cos-value*<br><br>**Example:**<br>`switch(config-pmap-c)# set cos 1` | Specifies the 802.1Q class of service (CoS) value. The range is from 0 to 7. The default value is 0. |
| Step 7 | **exit**<br><br>**Example:**<br>`switch(config-pmap-c)# exit`<br>`switch(config-pmap)#` | Exits policy map class configuration mode. |
| Step 8 | **exit**<br><br>**Example:**<br>`switch(config-pmap)# exit`<br>`switch(config)#` | Exits policy map configuration mode. |
| Step 9 | (Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*]<br><br>**Example:**<br>`switch(config)# show policy-map type control-plane` | Displays the control plane policy map configuration. |
| Step 10 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring the Control Plane Service Policy

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

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

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

**Before you begin**

Ensure that you have configured a control plane policy map.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal** <br> **Example:** <br> `switch# configure terminal` <br> `switch(config)#` | Enters global configuration mode. |
| **Step 2** | **control-plane** <br> **Example:** <br> `switch(config)# control-plane` <br> `switch(config-cp)#` | Enters control plane configuration mode. |
| **Step 3** | [**no**] **service-policy input** *policy-map-name* <br> **Example:** <br> `switch(config-cp)# service-policy input PolicyMapA` | Specifies a policy map for the input traffic. Repeat this step if you have more than one policy map. <br> You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds. |

|  | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **exit**<br><br>**Example:**<br><br>`switch(config-cp)# exit`<br>`switch(config)#` | Exits control plane configuration mode. |
| **Step 5** | (Optional) **show running-config copp** [**all**]<br><br>**Example:**<br><br>`switch(config)# show running-config copp` | Displays the CoPP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

**Related Topics**

# Configuring the CoPP Scale Factor Per Line Card

You can configure the CoPP scale factor per line card.

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

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **control-plane**<br><br>**Example:**<br><br>`switch(config)# control-plane`<br>`switch(config-cp)#` | Enters control plane configuration mode. |
| **Step 3** | **scale-factor** *value* **module** *multiple-module-range*<br><br>**Example:**<br><br>`switch(config-cp)# scale-factor 1.10`<br>`module 1-2` | Configures the policer rate per line card. The allowed scale factor value is from 0.10 to 2.00. When the scale factor value is configured, the policing values are multiplied by the corresponding scale factor value of the module, and it is programmed in the particular module. |

| | Command or Action | Purpose |
|---|---|---|
| | | To revert to the default scale factor value of 1.00, use the **no scale-factor** *value* **module** *multiple-module-range* command, or explicitly set the default scale factor value to 1.00 using the **scale-factor 1 module** *multiple-module-range* command. |
| **Step 4** | (Optional) **show policy-map interface control-plane**<br><br>**Example:**<br>`switch(config-cp)# show policy-map interface control-plane` | Displays the applied scale factor values when a CoPP policy is applied. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Changing or Reapplying the Default CoPP Policy

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

**Procedure**

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

**Related Topics**

# Copying the CoPP Best Practice Policy

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

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **copp copy profile** {**strict** \| **moderate** \| **lenient** \| **dense**} {**prefix** \| **suffix**} *string*<br><br>**Example:**<br>`switch# copp copy profile strict prefix abc` | Creates a copy of the CoPP best practice policy.<br><br>CoPP renames all class maps and policy maps with the specified prefix or suffix. |
| **Step 2** | (Optional) **show copp status**<br><br>**Example:**<br>`switch# show copp status` | Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the copied policy is not attached to the control plane. |
| **Step 3** | (Optional) **show running-config copp**<br><br>**Example:**<br>`switch# show running-config copp` | Displays the CoPP configuration in the running configuration, including the copied policy configuration. |

# Protocol ACL Filtering for Egress CoPP

The protocol ACL filtering for egress CoPP enables the NX-OS swtich to filter all traffic to control plane based on the host MAC, IPv4, and IPv6 address.

# Configuring ARP ACL Filtering for Egress CoPP

You can configure MAC ACL filtering at egress CoPP.

**Before you begin**

Ensure that you have configured a control plane policy map.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **hardware access-list tcam region erg-copp** *size* | Configures the size of the CoPP TCAM region. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`switch(config)# hardware access-list tcam region erg-copp 128` | |
| Step 3 | **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |
| Step 4 | **reload**<br>**Example:**<br>`switch(config)# reload` | Reloads the device.<br><br>**Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules. |
| Step 5 | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 6 | **mac access-list mac-foo-1**<br>**Example:**<br>`switch# mac access-list mac-foo-1`<br>`switch(config-mac-acl)#` | |
| Step 7 | **class-map type control-plane** [**match-all** \| **match-any**] *class-map-name*<br>**Example:**<br>`switch(config)# class-map type control-plane match-any c-map2`<br>`switch(config-cmap)#` | Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case-sensitive. |
| Step 8 | (Optional) **match access-group name** *access-list-name*<br>**Example:**<br>`switch(config-cmap)# match access-group name IP-foo-1` | |
| Step 9 | **policy-map type control-plane** *policy-map-name*<br>**Example:**<br>`switch(config)# policy-map type control-plane ClassMapA`<br>`switch(config-pmap)#` | Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case-sensitive. |
| Step 10 | **class** {*class-map-name* [**insert-before** *class-map-name2*] \| **class-default**}<br>**Example:** | Specifies a control plane class map name or the class default and enters control plane class configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-pmap)# class ClassMap2`<br>`switch(config-pmap-c)#` | The class-default class map is always at the end of the class map list for a policy map. |
| Step 11 | Enter one of the following commands:<br><br>• **police** [**cir**] {*cir-rate* [*rate-type*]}<br>• **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]<br>• **police** [**cir**] {*cir-rate* [*rate-type*]]}  **conform transmit** [**violate drop**]<br><br>**Example:**<br>`switch(config-pmap-c)# police cir 52000`<br>` bc 1000 packets` | Specifies the committed information rate (CIR). The rate range is as follows:<br><br>The committed burst (BC) range is as follows: |
| Step 12 | **control-plane Dynamic mode**<br><br>**Example:**<br>`switch(config)# control-plane dynamic`<br>`switch(config-cp-dyn)#` | Enters the control plane dynamic configuration mode. |
| Step 13 | **service-policy-dynamic input** *policy-map-name*<br><br>**Example:**<br>`switch(config-cp-dyn)#`<br>`service-policy-dynamic input PolicyMap1` | Specifies a policy map for the input traffic. |

# Configuring IP ACL Filtering for Egress CoPP

You can configure IP ACL filtering at egress CoPP.

### Before you begin

Ensure that you have configured a control plane policy map.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | [**no**] **hardware access-list tcam region erg-copp** *size*<br><br>**Example:**<br>`switch(config)# hardware access-list`<br>`tcam region erg-copp 128` | Configures the size of the egress CoPP TCAM region. |

| | | Command or Action | Purpose |
|---|---|---|---|
| Step 3 | | **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |
| Step 4 | | **reload**<br><br>**Example:**<br>`switch(config)# reload` | Reloads the device.<br><br>**Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules. |
| Step 5 | | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 6 | | **ip access-list IP-foo-1**<br><br>**Example:**<br>`switch# ip access-list mac-foo-1`<br>`switch(config-acl)#` | |
| Step 7 | | **permit tcp access-list IP-foo-1** *eq bgp*<br><br>**Example:**<br>`switch(config-acl)# 10 permit tcp`<br>`10.1.1.1/32 10.1.1.2/32 eq bgp` | |
| Step 8 | | **class-map type control-plane** [**match-all** \| **match-any**] *class-map-name*<br><br>**Example:**<br>`switch(config)# class-map type`<br>`control-plane match-any c-map2`<br>`switch(config-cmap)#` | Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive. |
| Step 9 | | **match access-group name** *access-list-name*<br><br>**Example:**<br>`switch(config-cmap)# match access-group`<br>` name IP-foo-1` | |
| Step 10 | | **policy-map type control-plane** *policy-map-name*<br><br>**Example:**<br>`switch(config)# policy-map type`<br>`control-plane ClassMapA`<br>`switch(config-pmap)#` | Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive. |

|        | **Command or Action** | **Purpose** |
|--------|----------------------|-------------|
| **Step 11** | **class** {*class-map-name* [**insert-before** *class-map-name2*] \| **class-default**} <br><br> **Example:** <br> `switch(config-pmap)# class ClassMap2` <br> `switch(config-pmap-c)#` | Specifies a control plane class map name or the class default and enters control plane class configuration mode. <br><br> The class-default class map is always at the end of the class map list for a policy map. |
| **Step 12** | Enter one of the following commands: <br><br> • **police** [**cir**] {*cir-rate* [*rate-type*]} <br> • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] <br> • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] <br><br> **Example:** <br> `switch(config-pmap-c)# police cir 52000 bc 1000 packets` <br><br> **Example:** <br> `switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes` | Specifies the committed information rate (CIR). The rate range is as follows: <br><br> The committed burst (BC) range is as follows: |
| **Step 13** | **control-plane Dynamic mode** <br><br> **Example:** <br> `switch(config)# control-plane dynamic` <br> `switch(config-cp-dyn)#` | Enters the control plane dynamic configuration mode. |
| **Step 14** | **service-policy-dynamic input** *policy-map-name* <br><br> **Example:** <br> `switch(config-cp-dyn)# service-policy-dynamic input PolicyMap1` | Specifies a policy map for the input traffic. ENd |

# Verifying the CoPP Configuration

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

| **Command** | **Purpose** |
|-------------|-------------|
| **show policy-map type control-plane** [**expand**] [**name** *policy-map-name*] | Displays the control plane policy map with associated class maps and CIR and BC values. |

| Command | Purpose |
|---------|---------|
| **show policy-map interface control-plane** | Displays the policy values with associated class maps and drops per policy or class map. It also displays the scale factor values when a CoPP policy is applied. When the scale factor value is the default (1.00), it is not displayed. <br><br> **Note**    The scale factor changes the CIR and BC values internally on each module, but the display shows the configured CIR and BC values only. The actual applied value on a module is the scale factor multiplied by the configured value. |
| **show class-map type control-plane** [*class-map-name*] | Displays the control plane class map configuration, including the ACLs that are bound to this class map. |
| **show copp diff profile** {**strict** \| **moderate** \| **lenient** \| **dense**} [**prior-ver**] **profile** {**strict** \| **moderate** \| **lenient** \| **dense**} **show copp diff profile** | Displays the difference between two CoPP best practice policies. <br><br> When you do not include the prior-ver option, this command displays the difference between two currently applied default CoPP best practice policies (such as the currently applied strict and currently applied moderate policies). <br><br> When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies). |
| **show copp profile** {**strict** \| **moderate** \| **lenient** \| **dense**} | Displays the details of the CoPP best practice policy, along with the classes and policer values. |

| Command | Purpose |
|---|---|
| **show running-config aclmgr** [**all**] | Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration. |
| **show running-config copp** [**all**] | Displays the CoPP configuration in the running configuration. |
| **show startup-config aclmgr** [**all**] | Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration. |

# Displaying the CoPP Configuration Status

### Procedure

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

### Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

# Monitoring CoPP

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **show policy-map interface control-plane** | Displays packet-level statistics for all classes that are part of the applied CoPP policy. Statistics are specified in terms of OutPackets (packets admitted to the control plane) and |

| Command or Action | Purpose |
|---|---|
| | DropPackets (packets dropped because of rate limiting). |

### Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

  Service-policy  input: copp-system-p-policy-strict

    class-map copp-system-p-class-critical (match-any)
      set cos 7
      police cir 19000 pps , bc 128 packets
      module 4 :
        transmitted 373977 packets;
        dropped 0 packets;
```

# Monitoring CoPP with SNMP

Beginning with Cisco Nexus Release 9.2(3), CoPP supports the Cisco class-based QoS MIB (cbQoSMIB). All CoPP elements can now be monitored (but not modified) using SNMP. This feature applies only to policies and their subelements (such as classes, match rules, and set actions) that are attached to the control plane. Elements of policies that are not in service on the control plane are not visible through SNMP.

The following cbQoSMIB tables are supported:

- ccbQosServicePolicy

- cbQosInterfacePolicy

- cbQosObjects

- cbQosPolicyMapCfg

- cbQosClassMapCfg

- cbQosMatchStmtCfg

- cbQosPoliceCfg

- cbQosSetCfg

> **Note** SNMP MIB is not supported for Dynamic CoPP.

# Clearing the CoPP Statistics

**Procedure**

|  | Command or Action | Purpose |
|---|---|---|
| **Step 1** | (Optional) switch# **show policy-map interface control-plane** | Displays the currently applied CoPP policy and per-class statistics. |
| **Step 2** | switch# **clear copp statistics** | Clears the CoPP statistics. |

**Example**

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

# Configuration Examples for CoPP

This section includes example CoPP configurations.

# CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-p-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-p-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-p-acl-arp
permit any any 0x0806

ip access-list copp-system-p-acl-tacas
permit udp any any eq 49

ip access-list copp-system-p-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-p-acl-icmp
permit icmp any any

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

class-map type control-plane match-any copp-system-p-class-normal
match access-group name copp-system-p-acl-icmp
match exception ip icmp redirect
```

```
match exception ip icmp unreachable
match exception ip option

policy-map type control-plane copp-system-p-policy

class copp-system-p-class-critical
police cir 19000 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-important
police cir 500 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-normal
police cir 300 pps bc 32 packets conform transmit violate drop

class class-default
police cir 50 pps bc 32 packets conform transmit violate drop

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

Create CoPP class and associate ACL:

```
class-map type control-plane copp-arp-class
match access-group name copp-arp-acl
```

Add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
class copp-arp-class
police pps 500
```

The following example shows to customize COPP limit:

```
copp copy profile strict suffix CUSTOMIZED-COPP
policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
class copp-class-redirect-CUSTOMIZED-COPP
police cir 1500 mbps bc 125 mbytes conform transmit violate drop
control-plane
service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

# Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup

        ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.


*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.


Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
```

```
Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no)[y]: <CR>

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n

  Enter the switch name : <CR>

  Enable license grace period? (yes/no) [n]: n

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway? (yes/no) [y]: n

  Configure advanced IP options? (yes/no) [n]: <CR>

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: <CR>

    Type of ssh key you would like to generate (dsa/rsa) : <CR>

  Configure the ntp server? (yes/no) [n]: n

  Configure default interface layer (L3/L2) [L3]: <CR>

  Configure default switchport interface state (shut/noshut) [shut]: <CR>

  Configure best practices CoPP profile (strict/moderate/lenient/dense/skip) [strict]:
strict


The following configuration will be applied:
  password strength-check
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-p-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

# Changing CoPP Policy limit

The following exampls shows to change CoPP limit to set PTP state stable across PTP interfaces.

```
copp copy profile strict suffix CUSTOMIZED-COPP
policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
class copp-class-redirect-CUSTOMIZED-COPP
police cir 1500 mbps bc 125 mbytes conform transmit violate drop
control-plane
service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

# Additional References for CoPP

This section provides additional information related to implementing CoPP.

**Related Documents**

| Related Topic | Document Title |
|---|---|
| Licensing | *Cisco NX-OS Licensing Guide* |

**Standards**

| Standards | Title |
|---|---|
| RFC 2698 | A Two Rate Three Color Marker |

# CHAPTER 26

# Configuring Rate Limits

This chapter describes how to configure rate limits for supervisor-bound traffic on Cisco NX-OS devices.

This chapter includes the following sections:

# About Rate Limits

Rate limits can prevent redirected packets for exceptions from overwhelming the supervisor module on a Cisco NX-OS device.

You can configure rate limits for the following types of redirected packets:

- Access-list log packets

- Bidirectional Forwarding Detection (BFD) packets

- Catch-all exception traffic

- Fabric Extender (FEX) traffic

- Layer 3 glean packets

- Layer 3 multicast data packets

- SPAN egress traffic

For Cisco Nexus 9200, 9332C, 9364C, 9300-EX, 9300-FX/FXP/FX2/FX3, and 9300-GX platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards, the CoPP policer rate is kilo bits per second. For other Cisco Nexus 9000 Series switches, the CoPP policer rate is in packets per second; However, it is kilo bits per second for SPAN egress traffic.

# Guidelines and Limitations for Rate Limits

Rate limits has the following configuration guidelines and limitations:

- You can set rate limits for supervisor-bound exception and redirected traffic. Use control plane policing (CoPP) for other types of supervisor-bound traffic.

  **Note** Hardware rate-limiters protect the supervisor CPU from excessive inbound traffic. The traffic rate allowed by the hardware rate-limiters is configured globally and applied to each individual I/O module. The resulting allowed rate depends on the number of I/O modules in the system. CoPP provides more granular supervisor CPU protection by utilizing the modular quality-of-service CLI (MQC).

- You can configure a hardware rate-limiter to show statistics for outbound traffic on SPAN egress ports. This rate-limiter is supported on all Cisco Nexus 9000, 9300, and 9500 Series switches, and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches.

- The rate-limiter on egress ports is limited per pipe on the Cisco Nexus 9300 and 9500 Series switches, Cisco Nexus 3164Q, 31128PQ, Cisco Nexus 3232C, and 3264Q switches. The rate-limiter on egress ports is limited per slice on the Cisco Nexus 9200 and 9300-EX Series switches.

- Cisco Nexus 9300 and 9500 Series switches, Cisco Nexus 3164Q, Cisco Nexus 31128PQ, Cisco Nexus 3232C, and Cisco Nexus 3264Q switches support both local and ERSPAN. However, the rate-limiter only applies to ERSPAN. You must configure e-racl ACL TCAM region to enable the rate-limiter on these switches. For more information, see the ACL TCAM Regions section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

- For Cisco Nexus 9200 and 9300-EX Series switches and the N9K-X9736C-EX, N9K-97160YC-EX, N9K-X9732C-EX, N9K-X9732C-EXM line cards, the SPAN egress rate-limiter applies to both ERSPAN and local SPAN. You do not require special TCAM carving to use the rate-limiter on these devices.

- For Cisco Nexus 92160YC-X, 92304QC, 9272Q, 9232C, 92300YC, 9348GC-FXP, 93108TC-FX, 93180YC-FX Series switches and Cisco Nexus 3232C and Cisco Nexus 3264Q switches, you should not configure both, sFlow and ERSPAN.

- Logging rate-limit is enabled by default. No default configuration is shown up in **show running-config** and in **show running-config all**. Use **show logging** cli to check if rate-limit is enabled. It has a dedicated field to verify if rate-limit is enabled or disabled.

  Once no logging rate-limit config is applied, it appears in the running-config and displayed in show logging output.

- The **rate-limit cpu direction** {**input** | **output** | **both**} **pps** *packets* **action log** command is not supported on Cisco Nexus 9000 Series switches.

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

# Default Settings for Rate Limits

This table lists the default settings for rate limits parameters.

*Table 47: Default Rate Limits Parameters Settings*

| Parameters | Default |
|---|---|
| Access-list log packets rate limit | 100 packets per second |
| BFD packets rate limit | 10000 packets per second |
| Exception packets rate limit | 50 packets per second |
| FEX packets rate limit | 12000 packets per second |
| Layer 3 glean packets rate limit | 100 packets per second |
| Layer 3 multicast data packets rate limit | 3000 packets per second |
| SPAN egress rate limit | No limit |
| R-L Class packets rate limit | 100 packets per second |
| SPAN packets rate limit | 50 packets per second |
| sFLOW packets rate limit | 40000 kilobits per second |
| VXLAN-OAM packets rate limit | 1000 packets per second |
| 100M-ethports packets rate limit | 10000 packets per second |
| SPAN egress disabled dot1x packets rate limit | 3000 packets per second |
| MPLS-OAM packets rate limit | 300 packets per second |
| Netflow packets rate limit | 120000 packets per second |
| SSX packets rate limit | 120000 packets per second |
| UCS-mgmt packets rate limit | 120000 packets per second |
| MDNS packets rate limit | 1024 packets per second |

# Configuring Rate Limits

You can set rate limits on supervisor-bound traffic.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **hardware rate-limiter access-list-log** {*packets* \| **disable**} [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter`<br>`access-list-log 200` | Configures rate limits for packets that are copied to the supervisor module for access list logging. The range is 0–10000. |
| **Step 3** | **hardware rate-limiter bfd** *packets* [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter`<br>`bfd 500` | Configures rate limits for bidirectional forwarding detection (BFD) packets. The range is 0–10000. |
| **Step 4** | **hardware rate-limiter exception** *packets* [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter`<br>`exception 500` | Configures rate limits for any exception traffic in the system that is not classified by the Control Plane Policing (CoPP) policy. The range is 0–10000. |
| **Step 5** | **hardware rate-limiter fex** *packets* [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter`<br>`fex 500` | Configures rate limits for supervisor-bound FEX traffic. The range is 0–10000. |
| **Step 6** | **hardware rate-limiter layer-3 glean** *packets* [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter`<br>`layer-3 glean 500` | Configures rate limits for Layer 3 glean packets. The range is 0–10000.<br><br>A node receiving traffic for a particular destination might be unable to forward traffic because it is unaware of the rewrite information or the physical layer interface behind which the destination resides. During this time, it is possible to install a glean entry in the data path for that destination. Because this might not be a pointer to the global punt adjacency, a reserved module or port value is used to punt such packets to the supervisor. This glean rate can be controlled using the given rate limiter. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** The CoPP policy controls the rate of glean packets that are forwarded to CPU due to hit of global punt adjacency. The Layer 3 glean hardware rate-limiter limits the number of glean packets that are redirected to CPU by sup-redirect acess-list. This is used in special cases such as, in the VXLAN environment when the packet is received from an unknown VTEP. |
| **Step 7** | **hardware rate-limiter layer-3 multicast local-groups** *packets* [**module** *module* [**port** *start end*]]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter layer-3 multicast local-groups 300` | Configures rate limits for Layer 3 multicast data packets that are punted for initiating a shortest-path tree (SPT) join. The range is 0–10000. |
| **Step 8** | **hardware rate-limiter span-egress** *rate* [**module** *module*]<br><br>**Example:**<br>`switch(config)# hardware rate-limiter span-egress 123` | Configures rate limits for SPAN for egress traffic. The range is 0–100000000.<br><br>**Note** You should not configure both sFlow and the SPAN egress rate-limiter. |
| **Step 9** | (Optional) **show hardware rate-limiter** [ **access-list-log \| bfd \|exception \| fex \| layer-3 glean \| layer-3 multicast local-groups \| \|module** *module*]<br><br>**Example:**<br>`switch# show hardware rate-limiter` | Displays the rate limit configuration. The module range is 1–30. |
| **Step 10** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Monitoring Rate Limits

You can monitor rate limits.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **show hardware rate-limiter** [**access-list-log** \| **bfd** \| **exception** \| **fex** \| **layer-3 glean** \| **layer-3 multicast local-groups** \| **span-egress** \| **module** *module*]<br><br>**Example:**<br><br>`switch# show hardware rate-limiter`<br>`access-list-log` | Displays the rate limit statistics. |

# Clearing the Rate Limit Statistics

You can clear the rate limit statistics.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **clear hardware rate-limiter** {**all** \| **access-list-log** \| **bfd** \| **exception** \| **fex** \| **layer-3 glean** \| **layer-3 multicast local-groups** \| **span-egress** [**module** *module*] }<br><br>**Example:**<br><br>`switch# clear hardware rate-limiter`<br>`access-list-log` | Clears the rate limit statistics. |

# Verifying the Rate Limit Configuration

To display the rate limit configuration information, perform the following tasks:

| Command | Purpose |
|---|---|
| **show hardware rate-limiter** [**access-list-log** \| **bfd** \| **exception** \| **fex** \| **layer-3 glean** \| **layer-3 multicast local-groups** \| **span-egress** \| **module** *module*] | Displays the rate limit configuration. |

# Configuration Examples for Rate Limits

The following example shows how to configure rate limits for packets copied to the supervisor module for access list logging:

```
switch(config)# hardware rate-limiter access-list-log
switch(config)# show hardware rate-limiter access-list-log
Units for Config: kilo bits per second
Allowed, Dropped & Total: aggregated since last clear counters
```

```
Module: 4
  R-L Class          Config          Allowed         Dropped           Total
  +----------------+--------+--------------+--------------+----------------
+
  access-list-log      100              0               0                0

  Port group with configuration same as default configuration
      Eth4/1-36

Module: 22
  R-L Class          Config          Allowed         Dropped           Total
  +----------------+--------+--------------+--------------+----------------
+
  access-list-log      100              0               0                0

  Port group with configuration same as default configuration
      Eth22/1-0
```

The following example shows how the SPAN egress rate limiter might be in conflict with sFlow:

```
switch(config)# hardware rate-limiter span-egress 123
Warning: This span-egress rate-limiter might affect functionality of sFlow
switch(config)# show hardware rate-limiter span-egress
Units for Config: kilo bits per second
Allowed, Dropped & Total: aggregated since Module: 1
R-L Class          Config          Allowed         Dropped          Total
+----------------+----------+--------------+--------------+----------------+
  L3 glean            100             0             0               0
  L3 mcast loc-grp   3000             0             0               0
  access-list-log     100             0             0               0
  bfd               10000             0             0               0
  exception            50             0             0               0
  fex                3000             0             0               0
  span                 50             0             0               0
  dpss               6400             0             0               0
  span-egress         123             0             0               0
<<configured
```

# Additional References for Rate Limits

This section includes additional information related to implementing rate limits.

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco NX-OS licensing | *Cisco NX-OS Licensing Guide* |

# Configuring MACsec

This document describes how to configure MACsec on Cisco NX-OS devices.

# About MACsec

Media Access Control Security (MACsec) an IEEE 802.1AE along with MACsec Key Agreement (MKA) protocol provide secure communications on Ethernet links. It offers the following :

- Provides line rate encryption capabilities.

- Helps to ensure data confidentiality by providing strong encryption at Layer 2.

- Provides integrity checking to help ensure that data cannot be modified in transit.

- Can be selectively enabled using a centralized policy to help ensure that it is enforced where required while allowing non-MACsec-capable components to access the network.

- Encrypts packets on a hop-by-hop basis at Layer 2, allowing the network to inspect, monitor, mark, and forward traffic according to your existing policies (unlike end-to-end Layer 3 encryption techniques that hide the contents of packets from the network devices they cross).

# Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the keychain after the lifetime is expired. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Configuring a MACsec Keychain and Keys, on page 617.

A key can roll over to a second key within the same keychain by configuring the second key (in the keychain) and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless (that is, the key rolls over without traffic interruption).

# Fallback Key

A MACsec session can fail due to a key/key name (CKN) mismatch or a finite key duration between the switch and a peer. If a MACsec session does fail, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

To configure a MACsec fallback key, see Configuring MACsec Fallback Key, on page 620.

# Licensing Requirements for MACsec

| Product | License Requirement |
|---------|---------------------|
| Cisco NX-OS | MACsec requires a Security license. For a complete explanation of the Cisco NX-OS licensing scheme to obtain and apply licenses, see the Cisco NX-OS Licensing Guide. |

# Guidelines and Limitations for MACsec

MACsec has the following guidelines and limitations:

- MACsec is supported on the following interface types:

  - Layer 2 switch ports (access and trunk)

  - Layer 3 routed interfaces (no subinterfaces)

**Note** Enabling MACsec on the Layer 3 routed interface also enables encryption on all the subinterfaces that are defined under that interface. However, selectively enabling MACsec on a subset of subinterfaces of the same Layer 3 routed interface is not supported.

- Layer 2 and Layer 3-port channels (no subinterfaces)

- Beginning with Cisco Nexus Release 10.2(1)F, Secure Channel Identifier (SCI) can be disabled from MACSec security tag (SecTAG) on Cisco Nexus 9000 ToR switches.

  - It is supported in FX2 and FX3 platforms.

  - It is supported in FX platforms with XPN cipher suites only

- When the Cisco Nexus ToR switches are downgraded from Cisco NX-OS Release 9.3.7 to Cisco NX-OS Release 9.3.6 and below releases, MACsec is not supported.

- MKA is the only supported key exchange protocol for MACsec. The Security Association Protocol (SAP) is not supported.

- Link-level flow control (LLFC) and priority flow control (PFC) are not supported with MACsec.

- Multiple MACsec peers (different SCI values) for the same interface are not supported.

- You can retain the MACsec configuration when you disable MACsec using the **macsec shutdown** command.

- MACsec sessions are liberal in accepting packets from a key server whose latest Rx and latest Tx flags have been retired after Tx SA installation for the first time. The MACsec session then converges into a secure state.

- Beginning with Cisco NX-OS Release 9.2(1), the following configurations are allowed:

  - Allowing MACSec policy to be modified while the policy is referenced by an interface.

  - Allowing different MACsec policies across different lanes of a breakout port.

- Beginning with Cisco Nexus Release 9.2(1), MACsec is supported on Cisco Nexus 93180YC-FX and Cisco Nexus 3264C-E switches.

- Beginning with Cisco Nexus Release 9.3(1), MACsec is supported on the Cisco Nexus 9364C, 9332C, and 9348GC-FXP switches. The following limitations are applicable when you use MACsec with these switches:

  - Cisco Nexus C9364C—MACsec is supported on 16 ports (Ports 49–64).

  - Cisco Nexus C9332C—MACsec is supported on 8 ports (Ports 25–32).

  - Cisco Nexus 9348GC-FXP—MACsec is supported on 6 ports (Ports 49–54).

| **Note** | On the Cisco Nexus 9364C, and 9332C platform switches, when MACsec is either configured or unconfigured on a port, there will be a port-flap occurrence irrespective of MACsec security-policy type. |
|---|---|

- Beginning with Cisco Nexus Release 9.3(1), you cannot apply MACsec configuration directly on port-channel interface. However, you can apply MACsec configurations directly on port-channel member ports. This applies to both NX-OS and vPC port-channels.

- Beginning with Cisco Nexus Release 9.3(3), MACsec is supported on Cisco Nexus 93216TC-FX2, Cisco Nexus 93360YC-FX2.

- Beginning with Cisco NX-OS Release 9.3(5), MACsec is supported on the following switches and line cards:

    - Cisco Nexus 93180YC-FX3S switches - MACsec is supported on all ports.

    - Cisco Nexus X9732C-FX, and X9788TC-FX line cards

- The N9K-X9736C-FX, N9K-X9732C-FX, N9K-C9348GC-FXP, N9K-C93180YC-FX, N9K-C93108TC-FX, N9K-X9788TC-FX, N9K-C9336C-FX2, N9K-C93240YC-FX2, N9K-C93216TC-FX2, N9K-C93360YC-FX2, N9K-C9364C, and N9K-C9332C cards and switches do not support MACsec on 1G ports. MACsec is not supported on any port on a mac block that has 1G ports on it.

- Beginning with Cisco NX-OS Release 10.1(1), the Cisco Nexus 93180YC-FX3, and 93108TC-FX3P switches support MACsec on all port speeds including 1G and 10G port speeds.

- MACsec is supported on Cisco Nexus 93240YC-FX2, 9336C-FX2, 93108TC-FX, 93180YC-FX switches and the X9736C-FX, and X9732C-EXM line cards.

- 1G is not supported on BV ports or retimer ports. For the retimer port details, see *Supported Retimer Ports* section of **Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.**

- Cisco Nexus 9000 Series switches do not support MACsec on any of the MACsec capable ports when QSA is being used.

    - Beginning with Cisco NX-OS Release 9.3(7), MACsec is supported by Cisco Nexus 9364C and 9336C-FX2 switches when QSA is being used.

    - Beginning with Cisco NX-OS Release 10.1(1), MACsec is supported by Cisco Nexus 9336C-FX2, 9336C-FX2-E, and 9364C switches when QSA is being used.

    - Beginning with Cisco NX-OS Release 10.1(2), MACsec is supported by Cisco Nexus 9300-FX3 platform switches when QSA is being used.

- Beginning with Cisco Nexus Release 10.1(1), MACsec is supported on Cisco Nexus 9336C-FX2-E.

- Beginning with Cisco Nexus Release 10.2(1)F, MACsec is supported on Cisco Nexus X9716D-GX.

- Beginning with Cisco NX-OS Release 10.2(1q)F, MACsec is supported on ports 25-32 of Cisco Nexus 9332D-GX2B switches.

- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec is supported on the Cisco Nexus N9K-C9348D-GX2A switches on 1-48 ports.

- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec supports Cisco Nexus X9736C-FX, and X9736Q-FX line cards with 10G QSA links.

- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec is supported on ports 1-16 of Cisco Nexus 9364D-GX2A switches.

- On the Cisco Nexus 9332D-GX2B, 9364D-GX2A and 9348D-GX2A switches and Cisco Nexus X9836DM-A line card, when MACsec is either configured or unconfigured on a port, a port-flap occurs irrespective of MACsec security-policy type.

- Beginning with Cisco NX-OS Release 10.3(1)F, MACsec is supported on Cisco Nexus X9836DM-A line card of Cisco Nexus 9800 platform switches.

- Beginning with Cisco NX-OS Release 10.3(2)F, MACsec is supported on Cisco Nexus 9408 switches with LEM modules X9400-16W and X9400-8D on all supported links.

- Beginning with Cisco Nexus Release 10.3(3)F, the cipher key enforcement feature provides the option to define the supported cipher suites from the most preferred to the least preferred on the Cisco Nexus 9332D-GX2B, 9336C-FX2, 93180YC-FX, and 93180YC-FX3 switches with following limitations:

  - Cipher Key Enforcement feature will work effectively only if it is prioritized as key server, else it will be in **init** or **pending** state of the session.

  - Cipher Key Enforcement feature is only supported for direct connections between 2 peers. If MKA session is with multiple peers, this feature is not expected to work properly.

  - During a peer cipher suite allowance change, session may not secure on the most preferred supported cipher suite.

  - When changing cipher from any to an enforced-peer cipher on a policy that is used on any secured MACsec session, it is recommended to flap the port after changing the cipher to reach expected behavior. If flapping is not done, the session shows secured on the switch while peer session shows pending on unsupported ciphers. It could also cause session to not get secured immediately even if supported cipher is present in enforce-peer cipher suites.

  - The Allowed Peer Cipher Suites (APSC) can not be empty or cannot have duplicates.

  - Cipher-suite and cipher-suite enforce-peer commands cannot coexist under the same policy.

  - While waiting for SAK Cipher-Enforcing Timer to timeout to try the next cipher suite in line, the data and control traffic might face one way traffic interruptions even with should secure mode. The interruption will only recover when session is secured.

- Beginning with Cisco Nexus Release 10.4(1)F, MACsec is supported on ports 49 to 54 of Cisco Nexus 9348GC-FX3 and 9348GC-FX3PH switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, MACsec is supported on all front panel ports (port 1 to 32) of Cisco Nexus 9332D-H2R platform switches. However, MACsec is not supported on Ethernet1/33 and Ethernet1/34.

- Beginning with Cisco Nexus Release 10.4(2)F, MACsec is supported on the below switches:

  - Cisco Nexus 93400LD-H1 on all ports.

  - Cisco Nexus 93108TC-FX3 on ports 49 to 54.

- Beginning with Cisco Nexus Release 10.4(3)F, MACsec is supported on the Cisco Nexus 9364C-H1 switches on ports 49 to 64.

- If the MACsec feature is configured, non-disruptive ISSU is not supported.

Keychain restrictions:

- You cannot overwrite the octet string for a MACsec key. Instead, you must create a new key or a new keychain.

- A new key in the keychain is configured when you enter **end** or **exit**. The default timeout for editor mode is 6 seconds. If the key is not configured with the key octet string or/and the send lifetime within the 6-second window, incomplete information may be used to bring up the MACsec session and could result in the session being stuck in an Authorization Pending state. If the MACsec sessions are not converged after the configuration is complete, you might be advised to shut/no shut the ports.

- For a given keychain, key activation times should overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.

- In addition to enabling the MACsec feature, a MACsec keychain must be configured on at least one interface to consume the security add-on license.

Fallback restrictions:

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and will show as rekeying on the old CA under status. And the MACsec session on the new key on primary PSK will be in init state.

- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.

- The key ID (CKN) used in the fallback key chain must not match any of the key IDs (CKNs) used in the primary key chain.

- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec policy restrictions:

- BPDU packets can be transmitted before a MACsec session becomes secure.

Layer 2 Tunneling Protocol (L2TP) restrictions:

- MACsec is not supported on ports configured for dot1q tunneling or L2TP.

- L2TP does not work if STP is enabled on trunk ports for non-native VLANs.

Statistics restrictions:

- Few CRC errors should occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).

- Secy statistics are cumulative and polled every 30 seconds.

- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.

- On Cisco Nexus 9300-FX3 platform switches, the **show macsec secy statistics** command supports rate statistics and the following rate related "CISCO-SECY-EXT-MIB" OIDs beginning with Cisco NX-OS Release 10.4(2)F.

  - cseSecyIfRxUncontrolledPktRate,

  - cseSecyIfRxControlledPktRate,

  - cseSecyIfTxUncontrolledPktRate,

  - cseSecyIfTxControlledPktRate

  - cseSecyIfRxControlledOctetRate

  - cseSecyIfTxControlledOctetRate

  - cseSecyIfRxUnControlledOctetRate

  - cseSecyIfTxUnControlledOctetRate

Interoperability restrictions:

- Interoperability of N9K-X9732C-EXM and other peer switches (other Cisco and non-Cisco switches) is supported only with the XPN cipher suite.

- MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. For interoperability between previous releases and Cisco NX-OS Release 9.2(1), you must use keys with the AES_256_CMAC cryptographic algorithm.

- For interoperability between previous releases and Cisco NX-OS Release 9.2(1), pad the MACsec key with zeros if it is less than 32 octets.

- On any Cisco NX-OS switch, you can configure only one unique combination of an alternate MAC address and Ethernet type on all interfaces.

- When using 1G optics on MACSEC capable module, it is recommended to change diagnostics mode to 'minimal'.

- When you attempt to downgrade from Cisco NX-OS Release 9.3(1) to a Cisco NX-OS release without per port channel member MACsec configuration support, when the switch has MACsec configurations on members of the same port channel interface that are different from each other, you may see the following error message:

```
Asymmetric macsec config is present on port-channel members. Please use
symmetric macsec config across members to perform Non-disruptive ISSU.
```

- Software support for MACsec and 50G is not available on the Cisco Nexus X9400-22L LEM card.

**EAPOL has the following guidelines and limitations:**

- In Cisco NX-OS Release 9.3(1), EAPOL configuration is not supported on Cisco Nexus 9332C and 9364C Series switches.

- Within the same slice of the forwarding engine, EAPOL ethertype and dot1q ethertype cannot have the same value.

- For enabling EAPOL configuration, the range of ethernet type between 0 to 0x599 is invalid.

- For enabling EAPOL configuration, on N9K-X9836DM-A line card, the only supported EAPOL mac addresses are range 0x0180c2000000 to 0x0180c20000ff.

- While configuring EAPOL packets, the following combinations must not be used:

  - Mac address 0100.0ccd.cdd0 with any ethertype

  - Any mac address with Ether types: 0xfff0, 0x800, 0x86dd

  - The default destination MAC address, 0180.c200.0003 with the default Ethernet type, 0x888e

  - Different EAPOL DMAC addresses on both MACsec peers. The MACsec session works only if the MACsec peer is sending MKAPDUs with the DMAC configured locally.

- Beginning with Cisco NX-OS Release 10.2(1)F, EAPOL is supported on Cisco Nexus 9300-FX3 Series switches.

# Enabling MACsec

Before you can access the MACsec and MKA commands, you must enable the MACsec feature.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **feature macsec**<br><br>**Example:**<br>`switch(config)# feature macsec` | Enables MACsec and MKA on the device. |
| Step 3 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. |

# Disabling MACsec

Beginning with Cisco NX-OS Release 9.2(1), disabling the MACsec feature only deactivates this feature and does not remove the associated MACsec configurations.

Disabling MACsec has the following conditions:

- MACsec shutdown is global command and is not available at the interface level.

- The macsec shutdown, show macsec mka session/summary, show macsec mka session detail, and show macsec mka/secy statistics commands will display the 'Macsec is shutdown' message. However, the show macsec policy and show key chain commands will display the output.

- Consecutive MACsec status changes from macsec shutdown to no macsec shutdown and vice versa needs a 30 seconds time interval in between the status change.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **macsec shutdown**<br><br>**Example:**<br><br>`switch(config)# macsec shutdown` | Disables the MACsec configuration on the device. The **no** option restores the MACsec feature. |
| **Step 3** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config`<br>`startup-config` | Copies the running configuration to the startup configuration. This step is required only if you want to retain the MACsec in the shutdown state after the switch reload. |

# Configuring a MACsec Keychain and Keys

You can create a MACsec keychain and keys on the device.

**Note**  Only MACsec keychains will result in converged MKA sessions.

**Before you begin**

Make sure that MACsec is enabled.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

| | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 2** | (Optional) [**no**] **key-chain macsec-psk no-show**<br><br>**Example:**<br>`switch(config)# key-chain macsec-psk no-show` | Hides the encrypted key octet string in the output of the **show running-config** and **show startup-config** commands by replacing the string with a wildcard character. By default, PSK keys are displayed in encrypted format and can be easily decrypted. This command applies only to MACsec keychains.<br><br>**Note** The octet string is also hidden when you save the configuration to a file. |
| **Step 3** | **key chain** *name* **macsec**<br><br>**Example:**<br>`switch(config)# key chain 1 macsec`<br>`switch(config-macseckeychain)#` | Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode. |
| **Step 4** | **key** *key-id*<br><br>**Example:**<br>`switch(config-macseckeychain)# key 1000`<br>`switch(config-macseckeychain-macseckey)#` | Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64.<br><br>**Note** The key must consist of an even number of characters. |
| **Step 5** | **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** \| **AES_256_CMAC**}<br><br>**Example:**<br>`switch(config-macseckeychain-macseckey)# key-octet-string abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm AES_256_CMAC` | Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command.<br><br>The key octet string includes the following:<br><br>• 0 Encryption Type - No encryption (default)<br><br>• 6 Encryption Type - Proprietary (Type-6 encrypted). For more information, see Enabling Type-6 Encryption on MACsec Keys, on page 519.<br><br>• 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. To interoperate between previous releases and Cisco NX-OS Release 7.0(3)I7(2) or a later release, you must use keys with the AES_256_CMAC cryptographic algorithm. |
| **Step 6** | **send-lifetime** *start-time* **duration** *duration*<br><br>**Example:**<br>`switch(config-macseckeychain-macseckey)# send-lifetime 00:00:00 Oct 04 2016 duration 100000` | Configures a send lifetime for the key. By default, the device treats the start time as UTC.<br><br>The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years). |
| **Step 7** | (Optional) **show key chain** *name*<br><br>**Example:**<br>`switch(config-macseckeychain-macseckey)# show key chain 1` | Displays the keychain configuration. |
| **Step 8** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-macseckeychain-macseckey)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# MACsec Packet-Number Exhaustion

Every MACsec frame contains a 32-bit packet number (PN), and it is unique for a given Security Association Key (SAK). Upon PN exhaustion (after reaching 75% of $2^{32}$- 1), SAK rekey takes place automatically to refresh the data plane keys and the PN will wrap around.

For example, on 10G full line rate @ 64 bytes, the SAK rekey will occur every 216 seconds due to PN exhaustion.

This is applicable when using GCM-AES-PN-128 or GCM-AES-PN-256 cipher-suites.

When GCM-AES-XPN-128 or GCM-AES-XPN-256 cipher-suite is used , the SAK rekey happens automatically when reaching 75% of $2^{64}$- 1 , which will take several years to exhaust the packet numbering. The cipher-suite is configurable under the macsec policy and the operational cipher-suite is determined by the key-server device.

It is recommended to use XPN ciphersuite on N9K-X9732C-EXM line card

# Configuring MACsec Fallback Key

Beginning with Cisco NX-OS Release 9.2(1), you can configure a fallback key on the device to initiate a backup session if the primary session fails as a result of a key/key name (CKN) mismatch or a finite key duration between the switch and peer.

**Before you begin**

Make sure that MACsec is enabled and a primary and fallback keychain and key ID are configured. See Configuring a MACsec Keychain and Keys.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters the global configuration mode. |
| **Step 2** | **interface** *name*<br><br>**Example:**<br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port. |
| **Step 3** | **macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name*<br><br>**Example:**<br>`switch(config-if)# macsec keychain kc2 policy abc fallback-keychain fb_kc2` | Specifies the fallback keychain to use after a MACsec session failure due to a key/key ID mismatch or a key expiration. The fallback key ID should not match any key ID from a primary keychain.<br><br>Fallback keychain configuration for each interface can be changed on the corresponding interface, without removing the MACsec configuration, by reissuing the same command with the fallback keychain name changed.<br><br>**Note** The command must be entered exactly the same as the existing configuration command for the interface, except for the fallback keychain name.<br><br>See Configuring a MACsec Keychain and Keys. |
| **Step 4** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Configuring a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

**Before you begin**

Make sure that MACsec is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **macsec policy** *name*<br><br>**Example:**<br><br>`switch(config)# macsec policy abc`<br>`switch(config-macsec-policy)#` | Creates a MACsec policy. |
| **Step 3** | **[no] cipher-suite** { { **enforce-peer** *<allowed-peer-cipher-suite1>* [*allowed-peer-cipher-suite2>* [*allowed-peer-cipher-suite3>* [*allowed-peer-cipher-suite4>*]]] } \| **<suite>**}<br><br>**Example:**<br><br>`switch(config-macsec-policy)#`<br>`cipher-suite enforce-peer`<br>`GCM-AES-XPN-256 GCM-AES-XPN-128` | Configures the sequence for the following cipher suites from the most preferred to the least preferred. The session gets secured on the most preferred cipher suite that is supported by the peer.: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256.<br><br>To unconfigure, you can either use the **no** form or overwrite the existing sequence with the required sequence preference.<br><br>**Note** • For this feature to work, ensure that the Cisco NX-OS switch is set as key server.<br><br>• If the peer supports cipher suites that are not included in the set of cipher suites defined in the **cipher-suite enforce-peer** command, the MKA session state will not be secured instead it will be pending. |
| **Step 4** | (Optional) **[no] include-sci**<br><br>**Example:** | Disables SCI in SecTAG. By default, SCI is always enabled. |

| | Command or Action | Purpose |
|---|---|---|
| | switch(config-macsec-policy)# no include-sci | **Note** To prevent packet drops, ensure that SCI tagging settings are consistent at both ingress and egress points. |
| Step 5 | **key-server-priority** *number*<br><br>**Example:**<br>switch(config-macsec-policy)# key-server-priority 0 | Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16. |
| Step 6 | **security-policy** *name*<br><br>**Example:**<br>switch(config-macsec-policy)# security-policy should-secure | Configures one of the following security policies to define the handling of data and control packets:<br><br>• **must-secure**—Packets not carrying MACsec headers will be dropped.<br><br>• **should-secure**—Packets not carrying MACsec headers will be permitted. This is the default value. |
| Step 7 | **window-size** *number*<br><br>**Example:**<br>switch(config-macsec-policy)# window-size 512 | Configures the replay protection window such that the secured interface will not accept any packet that is less than the configured window size. The range is from 0 to 596000000. |
| Step 8 | **sak-expiry-time** *time*<br><br>**Example:**<br>switch(config-macsec-policy)# sak-expiry-time 100 | Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0. |
| Step 9 | **conf-offset** *name*<br><br>**Example:**<br>switch(config-macsec-policy)# conf-offset CONF-OFFSET-0 | Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50. This command might be necessary for intermediate switches to use packet headers {dmac, smac, etype} like MPLS tags. |
| Step 10 | (Optional) **show macsec policy**<br><br>**Example:**<br>switch(config-macsec-policy)# show macsec policy | Displays the MACsec policy configuration. |
| Step 11 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br>switch(config-macsec-policy)# copy running-config startup-config | Copies the running configuration to the startup configuration. |

# Configuring MACsec EAP

Beginning with Cisco NX-OS Release 10.4(1)F, you can use MACsec EAP profile for 802.1X authentication.

**Before you begin**

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

- Configure MACsec command which specifies should-secure (default) or must-secure macsec policy

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet** *slot / port*<br><br>**Example:**<br>`switch(config)# interface ethernet 1/30`<br>`switch(config-if)#` | Selects the interface to configure and enters interface configuration mode. |
| **Step 3** | **[no] macsec eap policy** *policy name*<br><br>**Example:**<br>`switch(config-if)# macsec eap policy P1`<br>`switch(config-eap-profile)#` | Creates the MACsec eap profile.<br><br>The **no** form of the command is used to disable the MACsec eap profile. |
| **Step 4** | **[no] dot1x supplicant eap profile** *eap profile name* }<br><br>**Example:**<br>`switch(config-if)# dot1x supplicant eap profile` | Enters global configuration mode.<br><br>Configures the eap profile to be used by the supplicant. |

# QKD integration with SKIP on MACsec

### About Quantum-Safe Encryption

Recent advancements in quantum computing have exposed vulnerabilities in various cryptographic algorithms, making them unsecured for future applications. The RSA (integer factorization) and DHE (discrete logarithms) public key algorithms, which rely on computational complexity, are now at risk of being solved by quantum computers using Shor's or Grover's algorithm.

As a result, establishing a shared secret key between communicating parties has become a significant challenge. To avoid this issue, you can configure quantum-safe algorithms or implement a Quantum Key Distribution (QKD).

# About QKD Integration with Secure Key Integration Protocol

Integrating Secure Key Integration Protocol (SKIP) protocol to the switches empowers to establish communication with external quantum devices. This advancement allows for the utilization of Quantum Key Distribution (QKD) devices in the exchange of MACsec encryption keys between switches.

QKD operates on the principles of quantum physics, utilizing the quantum state of photons to encode and share information through an optical link. Additionally, an authenticated classical channel is used for sharing measurements. The change in quantum states helps the two end parties of the communication channel to identify any interception of their key.

QKD is a secured key exchange mechanism against quantum attacks even in the future advancements in cryptanalysis or quantum computing. QKD doesn't require continual updates based on discovered vulnerabilities.

# Guidelines and Limitations

The integration of QKD with SKIP for MACsec communication has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.4(3)F, Secure Key Integration Protocol protocol is supported on the following Cisco Nexus switches:

  - N9K-C9348GC-FXP

  - N9K-C93216TC-FX2

  - N9K-C93360YC-FX2

  - N9K-C9336C-FX2

  - N9K-C9348GC-FX3

  - N9K-C9348D-GX2A

  - N9K-C9332D-H2R

- You can use the SKIP protocol only in a point-to-point MACsec link encryption scenario.

- The SKIP protocol is available only on the interfaces that support MACsec encryption.

- Ensure that the QKD server is accessible through the management interface if switches have an HTTPS connection that is established with it.

- To establish a secure Transport Layer Security (TLS) connection and enable mutual authentication, you must install trustpoint certificates on the switch. These certificates allow the switch to obtain keys from the server. For more information, see chapter Configuring PKI in *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*

- MACsec PPK session and EAP-TLS sessions are not supported on the identical interface.

- If MACsec peers are connected to two different QKD servers, the QKD servers synchronize the keys to establish an MKA session. This synchronization ensures that the MACsec key (CKN) and key-string (CAK) are the same at both ends.

- QKD MACsec session fails due to configuration or invalid key errors. There is no FALLBACK mechanism for the preshared key (PSK).

# Configuring point-to-point MACsec Link Encryption Using SKIP

In point-to-point MACsec Link Encryption, secure encryption is established by using SKIP in switches. This encryption is set up between two interfaces in peer switches and requires the assistance of a QKD device network. Instead of the switches network, the QKD network shares the MACsec encryption key. Therefore, when a switch is required to create a MACsec link between peer switch interfaces, it contacts the external QKD device and requests the key. The external QKD device then generates a key pair consisting of the key ID and the key.

The Key ID acts as the unique ID string for the key (Shared Secret). The QKD device shares both the key ID and Key with the switch, while the switch only shares the key ID with its peer. The Peer switch uses this Key ID to retrieve encryption keys from its QKD device. Hence, Quantum networks always securely communicate encryption keys.

# Enabling Postquantum Cryptography

**Before you begin**

- Configure MACsec Pre-Shared Key (PSK).

- Configure MACsec in the PPK mode.

- An external QKD device network.

- Add the QKD server CA to the trustpoint in the switch and import the QKD server root CA certificate to the switch.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | switch# **configure terminal**<br><br>**Example:**<br><br>```<br>switch# configure terminal<br>switch(config)#<br>``` | Enters global configuration mode. |
| **Step 2** | switch(config)# **feature cryptopqc**<br><br>**Example:**<br><br>```<br>switch(config)# feature cryptopqc<br>``` | Enables post quantum cryptography (cryptopqc) on the switch. |
| **Step 3** | (Optional) switch(config)# **copy running-config startup-config**<br><br>**Example:**<br><br>```<br>switch(config)# copy running-config<br>startup-config<br>``` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

# Enabling MACsec and MKA features

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | switch# **configure terminal**<br><br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | switch(config)# **feature macsec**<br><br>**Example:**<br>`switch(config)# feature macsec` | Enables MACsec and MKA on the switch. |
| **Step 3** | (Optional) switch(config)# copy running-config startup-config<br><br>**Example:**<br>`switch(config)# copy running-config`<br>`startup-config` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

# Configuring Quantum Key Distribution Profile

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | switch (config)# **crypto qkd profile***name*<br><br>**Example:**<br>`switch(config)# crypto qkd profile ppk1` | Creates a QKD profile with name ppk1. |
| **Step 2** | switch (config)# **kme server**<*hostname*/*IP*> **port** *portnumber*<br><br>**Example:**<br>`switch(config-crypto-qkd-profile)# kme`<br>`server 172.0.0.2 port 6000` | Configures Key Management Engine (KME)/QKD server IP and TCP port number.<br><br>**Note**  Port number is optional. By default, port number will be 443. |
| **Step 3** | switch(config)#**transport tls authentication-type trustpoint**<*trustpoint name*><br><br>**Example:**<br>`switch(config-crypto-qkd-profile)#`<br>`transport tls authentication-type`<br>`trustpoint tp1` | Configures the CA (Certificate Authority) trustpoint. To create a trustpoint, refer to Configuring PKI section. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) switch(config)#**copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

# Enabling MACsec and MKA features

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config)# **macsec policy**<*name*><br><br>**Example:**<br><br>`switch(config)# [no] macsec policy test-policy` | Creates a MACsec policy. |
| **Step 2** | switch(config)# **ppk crypto-qkd-profile**<*name*><br><br>**Example:**<br><br>`switch(config-macsec-policy)# [no] ppk crypto-qkd-profile ppk1` | Configures the PPK profile name. |
| **Step 3** | (Optional) switch(config)# copy running-config startup-config<br><br>**Example:**<br><br>`switch(config)# copy running-config startup-config` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

# Configuration Examples

The following examples shows configuration of QKD profile and display of the configured details:

- Configuring QKD profile

```
switch(config)# feature cryptopqc
switch(config)#
switch(config)# crypto qkd profile ppk1
switch(config-crypto-qkd-profile)# kme server 168.20.1.2 port 5000
switch(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch(config-crypto-qkd-profile)# end
switch#
```

- Displaying QKD configuration

```
switch# show running-config cryptopqc
!Command: show running-config cryptopqc
!Running configuration last done at: Mon Jan 29 22:19:16 2024
!Time: Mon Jan 29 22:19:35 2024
version 10.4(3) Bios:version 05.51
```

```
feature cryptopqc
crypto qkd profile ppk1
kme server 168.20.1.2 port 5000
transport tls authentication-type trustpoint tp1
switch#
```

The following examples shows configuration of PPK profile on MACsec policy and display of the configured details:

- Configuring PPK profile on MACsec policy

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# macsec policy test
switch(config-macsec-policy) # ppk crypto-qkd-profile ppk1
switch(config-macsec-policy)# sak-expiry-time 60
switch(config-macsec-policy)# exit
switch(config)# end
```

- Display of configured MACsec policy

```
switch# show macsec policy test
MACSec Cipher     Pri Window Offset Security     SAKRekey   timeICV     Policy Indicator
      Include-SCI
-------------------------------- ---------------- ---- ------------ --------
-------------- -------------- ------------- ------------
test   GCM-AES-XPN-256 16     148809600 0          should-secure 60          FALSE
           TRUE
MACSec Policy     PPK Crypto-QKD-Profile Name
-------------------------------- --------------------------------
test              ppk1
switch#
```

The following example shows configuration of key chain, MACsec policy on an interface, and display of configured details:

- Configuring key chain

```
switch(config)# key chain KC1 macsec
switch(config-macseckeychain)#key 10100000
switch(config-macseckeychain-macseckey)#key-octet-string
F123456789ABCDEF0123456789ABCDEFF123456789ABCDEF0123456789ABCDEF cryptographic-algorithm
 AES_256_CMAC
switch(config-macseckeychain-macseckey)#exit
```

- Configuring MACsec policy to an interface

```
switch(config)# interface Ethernet 1/21
switch(config-if)# macsec keychain KC1 policy test
```

- Displaying MACsec session

```
switch(config)# show macsec mka session
Interface     Local-TxSCI #     Peers     Status     Key-Server Auth Mode
----------------- ------------------------------- ----------------- ------------------
 ----------------- ------------------
Ethernet1/21  6cb2.ae9f.e766/0001 1     Secured        No     PRIMARY-PPK
```

The following examples show configuring point-to-point MACsec QKD profile, binding QKD profile to MACsec policy and binding the MACsec policy to the interface:

> **Note** Make sure that the KME1 and KME2 servers must be active for connection through management port.

Configuring Swtich 1

```
switch1# configure terminal
switch1(config)# crypto ca trustpoint tp1
switch1config-trustpoint)# end
switch1#

switch1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)# feature cryptopqc
switch1(config)#
switch1(config)# crypto qkd profile PPK1
switch1(config-crypto-qkd-profile)# kme server KME1 port 7010
switch1(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch1(config-crypto-qkd-profile)# end
switch1#

switch1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)# feature macsec
switch1(config)#

switch1(config)# macsec policy MP1
switch1(config-macsec-policy)# ppk crypto-qkd-profile PPK1
switch1(config-macsec-policy)#exit
switch1(config-if)# interface Ethernet1/21
switch1(config-if)# macsec keychain KC1 policy MP1
switch1(config-if)#

switch1(config-if)# interface Ethernet1/22
switch1(config-if)# macsec keychain KC1 policy MP1
switch1(config-if)#
switch1(config-if)# end
switch1#
```

Configuring Swtich 2

```
switch2# configure terminal
switch2(config)# crypto ca trustpoint tp1
switch2config-trustpoint)# end
switch2#

switch2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)# feature cryptopqc
switch2(config)#
switch2(config)# crypto qkd profile PPK1
switch2(config-crypto-qkd-profile)# kme server KME2 port 7010
switch2(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch2(config-crypto-qkd-profile)#
switch2(config-crypto-qkd-profile)# end
switch2#

switch2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)# feature macsec
switch2(config)#
```

```
switch2(config)# macsec policy MP1
switch2(config-macsec-policy)# ppk crypto-qkd-profile PPK1
switch2(config-macsec-policy)# exit

switch2(config-if)# interface Ethernet1/53
switch2(config-if)# macsec keychain KC1 policy MP1
switch2(config-if)#
switch2(config-if)# interface Ethernet1/54
switch2(config-if)# macsec keychain KC1 policy MP1
switch2(config-if)# end
switch2#
```

The following shows the output of configuration on Switch 1 and Switch 2:

Switch 1:

```
switch1#
switch1# show macsec mka session
Interface      Local-TxSCI #      Peers      Status      Key-Server      Auth Mode
----------------- ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/22   3c8b.7ffe.0244/0001 1      Secured         Yes         PRIMARY-PPK
Ethernet1/21   3c8b.7ffe.0240/0001 1      Secured         Yes         PRIMARY-PPK
N9K3K STANDARD TEMPLATE FOR FEATURE REVIEWS
----------------- ------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
switch1#
```

Switch 2:

```
switch2#
switch2# show macsec mka session
Interface      Local-TxSCI #      Peers      Status      Key-Server      Auth Mode
----------------- ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/53   5451.deb8.62b4/0001 1      Secured         No          PRIMARY-PPK
Ethernet1/54   5451.deb8.62b8/0001 1      Secured         No          PRIMARY-PPK
----------------- ------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
switch2#
```

# About Configurable EAPOL Destination and Ethernet Type

Beginning Cisco NX-OS Release 9.2(2), Cisco enables networks with WAN MACsec to change the Extensible Authentication Protocol (EAP) over LAN (EAPOL) protocol destination address, and the Ethernet type values to nonstandard values.

Configurable EAPOL MAC and Ethernet type provides you the ability to change the MAC address and the Ethernet type of the MKA packet, inorder to allow CE device to form MKA sessions over the ethernet networks that consume the standard MKA packets.

The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

This feature is available at the interface level and the alternate EAPOL configuration can be changed on any interface at any given time as follows:

- If the MACsec is already configured on an interface, the sessions will come up with a new alternate EAPOL configuration.

- When MACsec is not configured on an interface, the EAPOL configuration is applied to the interface and is effective when MACsec is configured on that inferface.

# Enabling EAPOL Configuration

You can enable the EAPOL configuration on any available interface.

**Before you begin**

Make sure that MACsec is enabled.

**Procedure**

|  | **Command or Action** | **Purpose** |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface** *name*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port. |
| **Step 3** | **eapol mac-address** *mac_address* [**ethertype** *eth_type*] | Enables the EAPOL configuration on the specified interface type and identity.<br><br>**Note**     If the ethernet type is not specified, the default ethernet type of MKA packets, which is 0x888e, is considered. |
| **Step 4** | **eapol mac-address broadcast-address** [**ethertype** *eth_type*] | Enables the broadcast address as the alternate mac address. |
| **Step 5** | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-macseckeychain-macseckey)#`<br>` copy running-config startup-config` | Copies the running configuration to the startup configuration. |
| **Step 6** | **show macsec mka session detail** | Displays the EAPOL settings. |

# Disabling EAPOL Configuration

You can disable the EAPOL configuration on any available interface.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **configure terminal**<br><br>**Example:**<br><br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| Step 2 | **interface** *name*<br><br>**Example:**<br><br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port. |
| Step 3 | [**no**] **eapol mac-address** *mac_address* [**ethertype** *eth_type*] | Disables the EAPOL configuration on the specified interface type and identity. |
| Step 4 | (Optional) **copy running-config startup-config**<br><br>**Example:**<br><br>`switch(config-macseckeychain-macseckey)#`<br>` copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Verifying the MACsec Configuration

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

| Command | Purpose |
|---|---|
| **show key chain** *name* | Displays the keychain configuration. |
| **show macsec mka session** [**interface** *type slot*/*port*] [**detail**] | Displays information about the MACsec MKA session for a specific interface or for all interfaces. |
| **show macsec mka session details** | Displays information about the MAC address and the ethernet type that is currently used by the interfaces for all EAPOL packets. |
| **show macsec mka summary** | Displays the MACsec MKA configuration. |
| **show macsec policy** [*policy-name*] | Displays the configuration for a specific MACsec policy or for all MACsec policies. |
| **show running-config macsec** | Displays the running configuration information for MACsec. |

The following example displays information about the MACsec MKA session for all interfaces. .

```
switch# show macsec mka session
Interface         Local-TxSCI                        #Peers            Status
Key-Server        Auth Mode
----------------- -------------------------------- ------------------ -----------------
----------------- ------------------
Ethernet2/2       2c33.11b8.7d14/0001                1                 Secured
Yes               PRIMARY-PSK
Ethernet2/3       2c33.11b8.7d18/0001                1                 Secured
```

```
Yes                PRIMARY-PSK
----------------- ------------------------------ ----------------- -----------------
----------------- -----------------
Total Number of Sessions : 2
        Secured Sessions : 2
        Pending Sessions : 0
```

The following example displays information about the MACsec MKA session for a specific interface. In addition to the common elements of the table as described in the previous example, the following also identifies the authentication mode which defines the current MACsec session type.

```
switch# show macsec mka session interface ethernet 1/1

Interface      Local-TxSCI         # Peers  Status       Key-Server    Auth Mode
-------------- ------------------- --------- ------------ ------------- -------------
Ethernet1/1    70df.2fdc.baf4/0001      0    Pending      Yes           PRIMARY-PSK
Ethernet1/1    70df.2fdc.baf4/0001      1    Secured      No            FALLBACK-PSK
```

The following example displays detailed information about the MACsec MKA session for a specific Ethernet interface:

```
Interface Name          : Ethernet2/2
    Session Status                    : SECURED - Secured MKA Session with MACsec
    Local Tx-SCI                      : 2c33.11b8.7d14/0001
    Local Tx-SSCI                     : 2
    MKA Port Identifier               : 2
    CAK Name (CKN)                    : 12
    CA Authentication Mode            : PRIMARY-PSK
    Member Identifier (MI)            : B54263EF7949A561E25CE617
    Message Number (MN)               : 523
    MKA Policy Name                   : tests2
    Key Server Priority               : 16
    Key Server                        : Yes
    Include ICV                       : No
    SAK Cipher Suite                  : GCM-AES-XPN-256
    SAK Cipher Suite (Operational)    : GCM-AES-XPN-256
    Replay Window Size                : 148809600
    Confidentiality Offset            : CONF-OFFSET-0
    Confidentiality Offset (Operational): CONF-OFFSET-0
    Latest SAK Status                 : Rx & TX
    Latest SAK AN                     : 0
    Latest SAK KI                     : B54263EF7949A561E25CE61700000001
    Latest SAK KN                     : 1
    Last SAK key time                 : 12:59:38 PST Tue Mar 19 2019
    CA Peer Count                     : 1
    Eapol dest mac                    : 0180.c200.0003
    Ether-type                        : 0x888e
Peer Status:
    Peer MI                           : 2C2C090E62A96F4D6E018210
    RxSCI                             : 2c33.11b8.8b88/0001
    Peer CAK                          : Match
    Latest Rx MKPDU                   : 13:16:54 PST Tue Mar 19 2019
```

The following example displays the MACsec MKA configuration:

```
switch# show macsec mka summary
Interface        MACSEC-policy                   Keychain
----------------- ------------------------------ -----------------------
Ethernet2/13     1                                1/10000000000000000
Ethernet2/14     1                                1/10000000000000000
```

The following example displays the configuration for all MACsec policies:

```
switch# show macsec policy
MACSec Policy        Cipher         Pri  Window    Offset   Security   SAK Rekey time ICV
 Indicator  Include-SCI
------------------ -------------    --- --------   -------- --------   --------------
------------- -------------
KC256-Po117b       GCM-AES-256    16   148809600   0       should-secure  pn-rollover
FALSE        True
pol1               GCM-AES-XPN-256 100  148809600  30       must-secure    60
FALSE        True
pol256-FanO        GCM-AES-XPN-256 16   148809600   0       must-secure    60
FALSE        True
pol256-MCT         GCM-AES-XPN-256 16   148809600   0       should-secure  60
FALSE        FALSE
system-default-
macsec-policy      GCM-AES-XPN-256 16   148809600   0       should-secure  pn-rollover
FALSE        FALSE
test1              GCM-AES-XPN-256 16   148809600   0       should-secure  pn-rollover
FALSE        True
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is not configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 075e701e1c5a4a5143475e5a527d7c7c706a6c724306170103555a5c57510b051e47080
a05000101005e0e50510f005c4b5f5d0b5b070e234e4d0a1d0112175b5e cryptographic-algorithm
AES_256_CMAC
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

# Displaying MACsec Statistics

You can display MACsec statistics using the following commands.

| Command | Description |
|---------|-------------|
| **show macsec mka statistics** [**interface** *type slot*/*port*] | Displays MACsec MKA statistics. |
| **show macsec secy statistics** [**interface** *type slot*/*port*] | Displays MACsec security statistics. |

The following example shows the MACsec MKA statistics for a specific Ethernet interface:

```
switch# show macsec mka statistics interface ethernet 2/2

Per-CA MKA Statistics for Session on interface (Ethernet2/2) with CKN 0x10
================================================================================
CA Statistics
   Pairwise CAK Rekeys..... 0

SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
```

```
        SAKs Received.......... 0
        SAK Responses Received.. 0

    MKPDU Statistics
        MKPDUs Transmitted...... 1096
            "Distributed SAK".. 0

        MKPDUs Validated & Rx... 0
            "Distributed SAK".. 0

    MKA Statistics for Session on interface (Ethernet2/2)
    ========================================================
    CA Statistics
        Pairwise CAK Rekeys..... 0

    SA Statistics
        SAKs Generated......... 0
        SAKs Rekeyed........... 0
        SAKs Received.......... 0
        SAK Responses Received.. 0

    MKPDU Statistics
        MKPDUs Transmitted...... 1096
            "Distributed SAK".. 0
        MKPDUs Validated & Rx... 0
            "Distributed SAK".. 0
        MKPDUs Tx Success......... 1096
        MKPDUs Tx Fail............ 0
        MKPDUS Tx Pkt build fail... 0
        MKPDUS No Tx on intf down.. 0
        MKPDUS No Rx on intf down.. 0
        MKPDUs Rx CA Not found..... 0
        MKPDUs Rx Error........... 0
        MKPDUs Rx Success......... 0

    MKPDU Failures
        MKPDU Rx Validation ..................... 0
        MKPDU Rx Bad Peer MN.................... 0
        MKPDU Rx Non-recent Peerlist MN.......... 0
        MKPDU Rx Drop SAKUSE, KN mismatch........ 0
        MKPDU Rx Drop SAKUSE, Rx Not Set......... 0
        MKPDU Rx Drop SAKUSE, Key MI mismatch.... 0
        MKPDU Rx Drop SAKUSE, AN Not in Use...... 0
        MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set... 0
        MKPDU Rx Drop Packet, Ethertype Mismatch. 0

    SAK Failures
        SAK Generation................... 0
        Hash Key Generation.............. 0
        SAK Encryption/Wrap.............. 0
        SAK Decryption/Unwrap............ 0

    CA Failures
        ICK Derivation................... 0
        KEK Derivation................... 0
        Invalid Peer MACsec Capability... 0

    MACsec Failures
        Rx SA Installation............... 0
        Tx SA Installation............... 0
```

The following example shows the MACsec security statistics for a specific Ethernet interface.

**Note** The following differences exist for uncontrolled and controlled packets in Rx and Tx statistics:

- Rx statistics:
  - Uncontrolled = Encrypted and unencrypted
  - Controlled = Decrypted

- Tx statistics:
  - Uncontrolled = Unencrypted
  - Controlled = Encrypted
  - Common = Encrypted and unencrypted

```
switch(config)# show macsec secy statistics interface e2/28/1

Interface Ethernet2/28/1 MACSEC SecY Statistics:
-------------------------------------------
Interface Rx Statistics:
   Unicast Uncontrolled Pkts: 14987
   Multicast Uncontrolled Pkts: 1190444
   Broadcast Uncontrolled Pkts: 4
   Uncontrolled Pkts - Rx Drop: 0
   Uncontrolled Pkts - Rx Error: 0
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 247583
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   In-Octets Uncontrolled: 169853963 bytes
   In-Octets Controlled: 55027017 bytes
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)

Interface Tx Statistics:
   Unicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 205429
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Out-Octets Uncontrolled: N/A (N9K-X9736C-FX not supported)
   Out-Octets Controlled: 20612648 bytes
   Out-Octets Common: 151787484 bytes
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
```

```
        SECY Rx Statistics:
          Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
            Control Pkts: 952284
            Untagged Pkts: N/A (N9K-X9736C-FX not supported)
            No Tag Pkts: 0
            Bad Tag Pkts: 0
            No SCI Pkts: 0
            Unknown SCI Pkts: 0
            Tagged Control Pkts: N/A (N9K-X9736C-FX not supported)

        SECY Tx Statistics:
            Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
            Control Pkts: 967904
            Untagged Pkts: N/A (N9K-X9736C-FX not supported)

        SAK Rx Statistics for AN [3]:
            Unchecked Pkts: 0
            Delayed Pkts: 0
            Late Pkts: 0
            OK Pkts: 1
            Invalid Pkts: 0
            Not Valid Pkts: 0
            Not-Using-SA Pkts: 0
            Unused-SA Pkts: 0
            Decrypted In-Octets: 235 bytes
            Validated In-Octets: 0 bytes

        SAK Tx Statistics for AN [3]:
            Encrypted Protected Pkts: 2
            Too Long Pkts: N/A (N9K-X9736C-FX not supported)
            SA-not-in-use Pkts: N/A (N9K-X9736C-FX not supported)
            Encrypted Protected Out-Octets: 334 bytes
switch(config)#
```

# Configuration Example for MACsec

The following example shows how to configure a user-defined MACsec policy and then apply the policy to interfaces:

```
switch(config)# macsec policy 1
switch(config-macsec-policy)# cipher-suite GCM-AES-256
switch(config-macsec-policy)# window-size 512
switch(config-macsec-policy)# key-server-priority 0
switch(config-macsec-policy)# conf-offset CONF-OFFSET-0
switch(config-macsec-policy)# security-policy should-secure
switch(config-macsec-policy)# exit

switch(config)# int e2/13-14
switch(config-if-range)# macsec keychain 1 policy 1
switch(config-if-range)# exit
switch(config)# show macsec mka summary
Interface          MACSEC-policy                      Keychain
------------------ -------------------------------- ------------------------
Ethernet2/13       1                                  1/10000000000000000
Ethernet2/14       1                                  1/10000000000000000

switch(config)# show macsec mka session
Interface       Local-TxSCI         # Peers    Status     Key-Server
-------------- ------------------- ---------- ---------- -------------
Ethernet2/13   006b.f1be.d31c/0001  1          Secured    Yes
Ethernet2/14   006b.f1be.d320/0001  1          Secured    No
```

```
switch(config)# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:53:40 2016

version 9.2(1)feature macsec
macsec policy 1
  cipher-suite GCM-AES-256
  key-server-priority 0
  window-size 512
  conf-offset CONF-OFFSET-0
  security-policy should-secure

interface Ethernet2/13
  macsec keychain 1 policy 1

interface Ethernet2/14
  macsec keychain 1 policy 1
```

The following example shows how to configure a MACsec keychain and then add the system default MACsec policy to the interfaces:

```
switch(config)# key chain 1 macsec
switch(config-macseckeychain)# key 1000
switch(config-macseckeychain-macseckey)# key-octet-string
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm
aes_256_CMAC
switch(config-macseckeychain-macseckey)# exit

switch(config)# int e2/13-14
switch(config-if-range)# macsec keychain 1
switch(config-if-range)# exit
switch(config)#

switch(config)# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:50:16 2016
version 7.0(3)I4(5)
feature macsec
interface Ethernet2/13
  macsec keychain 1 policy system-default-macsec-policy
interface Ethernet2/14
  macsec keychain 1 policy system-default-macsec-policy

switch(config)# show macsec mka session
Interface       Local-TxSCI                     # Peers           Status
Key-Server      Auth Mode
----------------- ------------------------------- ----------------- -----------------
----------------- ------------------
Ethernet2/2     2c33.11b8.7d14/0001             1                 Secured
Yes             PRIMARY-PSK
Ethernet2/3     2c33.11b8.7d18/0001             1                 Secured
Yes             PRIMARY-PSK
----------------- ------------------------------- ----------------- -----------------
----------------- ------------------
Total Number of Sessions : 2
        Secured Sessions : 2
        Pending Sessions : 0

switch(config)# show macsec mka summary
Interface       Status  Cipher (Operational)  Key-Server  MACSEC-policy  Keychain
Fallback-keychain
----------------- -------- --------------------- ------------ -------------- ----------
---------------
Ethernet2/1     down      -                     -           tests1         keych1
```

```
 no keychain
Ethernet2/2        Secured  GCM-AES-XPN-256      Yes        tests2      keych2
 no keychain
Ethernet2/3        Secured  GCM-AES-256          Yes        tests3      keyc3
 no keychain
```

The following example shows the configuration and output of Peer Enforce Cipher configuration feature MACsec:

```
switch# show key chain
Key-Chain KC1 Macsec
Key 10000000 -- text 7
"0729701e1d5d4c53404a522d26090f010e63647040534355560e007971772a263e30080a0407070303530227257b73213556550958525a771b1650382734362e2a"
cryptographic-algorithm AES_256_CMAC
send lifetime (always valid) [active]

Key-Chain KC2 Macsec
Key 10100000 -- text 7
"0729701e1d5d4c53404a522d26090f010e63647040534355560e007971772a263e30080a0407070303530227257b73213556550958525a771b1650382734362e2a"
cryptographic-algorithm AES_256_CMAC
send lifetime (always valid) [active]

switch#
switch# show run macsec

!Command: show running-config macsec
!Running configuration last done at: Mon Apr 17 16:49:57 2023
!Time: Mon Apr 17 16:50:09 2023

version 10.3(3) Bios:version 05.47
feature macsec

macsec policy MP1
cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128
macsec policy MP2
cipher-suite enforce-peer GCM-AES-256
macsec policy Test-MP1
cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128
macsec policy Test-MP2
cipher-suite enforce-peer GCM-AES-256


interface Ethernet1/97/1
macsec keychain KC1 policy MP1

interface Ethernet1/97/2
macsec keychain KC2 policy MP2

switch#

switch# show macsec policy
MACSec Policy Cipher Pri Window Offset Security SAK Rekey time ICV Indicator Include-SCI
-------------------------------- ---------------- ---- ------------ -------- --------------
 -------------- ------------- -------------
MP1 Enforce-Peer 16 148809600 0 should-secure pn-rollover FALSE TRUE
MP2 Enforce-Peer 16 148809600 0 should-secure pn-rollover FALSE TRUE
system-default-macsec-policy GCM-AES-XPN-256 16 148809600 0 should-secure pn-rollover FALSE
 TRUE

MACSec Policy Cipher-Suite Enforce-Peer
-------------------------------- -----------------------------------------------------
MP1 GCM-AES-XPN-256 GCM-AES-XPN-128
```

```
MP2 GCM-AES-256
switch#
```

The following example shows the sample output of the **show macsec mka session detail** command:

```
switch# show macsec mka session details
Detailed Status for MKA Session
-----------------------------------
Interface Name : Ethernet1/97/1
Session Status : SECURED - Secured MKA Session with MACsec
Local Tx-SCI : c4f7.d530.1484/0001
Local Tx-SSCI : 1
MKA Port Identifier : 1
CAK Name (CKN) : 10000000
CA Authentication Mode : PRIMARY-PSK
Member Identifier (MI) : D94B90E3FDB111CE583E7158
Message Number (MN) : 111
MKA Policy Name : MP1
Key Server Priority : 16
Key Server : Yes
Include ICV : No
SAK Cipher Suite : GCM-AES-XPN-128
SAK Cipher Suite (Operational) : GCM-AES-XPN-128
Replay Window Size : 148809600
Confidentiality Offset : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status : Rx & TX
Latest SAK AN : 1
Latest SAK KI : D94B90E3FDB111CE583E715800000001
Latest SAK KN : 1
Last SAK key time : 16:48:41 PST Mon Apr 17 2023
CA Peer Count : 1
Eapol dest mac : 0180.c200.0003
Ether-type : 0x888e
Peer Status:
Peer MI : 00110000000100010000000001
RxSCI : 0011.0000.0001/0001
Peer CAK : Match
Latest Rx MKPDU : 16:52:07 PST Mon Apr 17 2023

Interface Name : Ethernet1/97/2
Session Status : SECURED - Secured MKA Session with MACsec
Local Tx-SCI : c4f7.d530.1485/0001
Local Tx-SSCI : 1
MKA Port Identifier : 1
CAK Name (CKN) : 10100000
CA Authentication Mode : PRIMARY-PSK
Member Identifier (MI) : 43AE54C19982238C298E0241
Message Number (MN) : 107
MKA Policy Name : MP2
Key Server Priority : 16
Key Server : Yes
Include ICV : No
SAK Cipher Suite : GCM-AES-256
SAK Cipher Suite (Operational) : GCM-AES-256
Replay Window Size : 148809600
Confidentiality Offset : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status : Rx & TX
Latest SAK AN : 0
Latest SAK KI : 43AE54C19982238C298E024100000001
Latest SAK KN : 1
Last SAK key time : 16:48:42 PST Mon Apr 17 2023
CA Peer Count : 1
Eapol dest mac : 0180.c200.0003
```

```
Ether-type : 0x888e
Peer Status:
Peer MI : 00270000001000100000001
RxSCI : 0027.0000.0001/0001
Peer CAK : Match
Latest Rx MKPDU : 16:52:06 PST Mon Apr 17 2023
switch#
```

# XML Examples

MACsec supports XML output for the following **show** commands for scripting purposes using | **xml**:

- **show key chain** *name* | **xml**

- **show macsec mka session interface** *interface slot/port* **details** | **xml**

- **show macsec mka statistics interface** *interface slot/port* | **xml**

- **show macsec mka summary** | **xml**

- **show macsec policy** *name* | **xml**

- **show macsec secy statistics interface** *interface slot/port* | **xml**

- **show running-config macsec** | **xml**

The following are example outputs for each of the preceding **show** commands:

Example 1: Displays the keychain configuration.

```
switch# show key chain "Kc2" | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0:rpm">
 <nf:data>
  <show>
   <key>
    <chain>
     <__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
      <keychain>Kc2</keychain>
     </__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
    </chain>
   </key>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 2: Displays information about the MACsec MKA session for a specific interface.

```
switch# show macsec mka session interface ethernet 4/31 details | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <mka>
     <session>
      <__XML__OPT_Cmd_show_macsec_mka_session_interface>
       <interface>
```

```
                        <__XML__INTF_ifname>
                         <__XML__PARAM_value>
                          <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
                         </__XML__PARAM_value>
                        </__XML__INTF_ifname>
                      </interface>
                      <__XML__OPT_Cmd_show_macsec_mka_session_details>
                       <details/>
                       <__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
                        <__readonly__>
                         <TABLE_mka_session_details>
                          <ROW_mka_session_details>
                           <ifname>Ethernet4/31</ifname>
                           <status>Secured</status>
                           <sci>0c75.bd03.5360/0001</sci>
                           <ssci>1</ssci>
                           <port_id>1</port_id>
                           <mi>020000000000000000000000000000000000000000000000000000000000000000
</mi>
                           <mi>F511280A765CE41C79458753</mi>
                           <mn>2770</mn>
                           <policy>am2</policy>
                           <ks_prio>0</ks_prio>
                           <keyserver>No</keyserver>
                           <cipher>GCM-AES-XPN-256</cipher>
                           <window>512</window>
                           <conf_offset>CONF-OFFSET-0</conf_offset>
                           <sak_status>Rx &amp; TX</sak_status>
                           <sak_an>1</sak_an>
                           <sak_ki>516486241</sak_ki>
                           <sak_kn>90</sak_kn>
                           <last_sak_rekey_time>07:12:02 UTC Fri Jan 20 2017</last_sak_rekey_ti
me>
                          </ROW_mka_session_details>
                         </TABLE_mka_session_details>
                        </__readonly__>
                       </__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
                      </__XML__OPT_Cmd_show_macsec_mka_session_details>
                     </__XML__OPT_Cmd_show_macsec_mka_session_interface>
                    </session>
                   </mka>
                  </macsec>
                 </show>
                </nf:data>
               </nf:rpc-reply>
              ]]>]]>
```

Example 3: Displays MACsec MKA statistics.

```
switch# show macsec mka statistics interface ethernet 4/31 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
<macsec>
     <mka>
      <statistics>
       <__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
        <interface>
         <__XML__INTF_ifname>
          <__XML__PARAM_value>
           <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
           <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
```

```
        </__XML__PARAM_value>
       </__XML__INTF_ifname>
      </interface>
      <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
       <__readonly__>
        <TABLE_mka_intf_stats>
         <ROW_mka_intf_stats>
          <TABLE_ca_stats>
           <ROW_ca_stats>
            <ca_stat_ckn>0x2</ca_stat_ckn>
            <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
            <sa_stat_sak_generated>0</sa_stat_sak_generated>
            <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
            <sa_stat_sak_received>91</sa_stat_sak_received>
            <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
            <mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
            <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
            <mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
            <mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
           </ROW_ca_stats>
          </TABLE_ca_stats>
         </ROW_mka_intf_stats>
        </TABLE_mka_intf_stats>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
      <interface>
       <__XML__INTF_ifname>
        <__XML__PARAM_value>
         <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
        </__XML__PARAM_value>
       </__XML__INTF_ifname>
      </interface>
      <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
       <__readonly__>
        <TABLE_mka_intf_stats>
         <ROW_mka_intf_stats>
          <TABLE_idb_stats>
           <ROW_idb_stats>
            <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
            <sa_stat_sak_generated>0</sa_stat_sak_generated>
            <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
            <sa_stat_sak_received>91</sa_stat_sak_received>
            <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
            <mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
            <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
            <mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
            <mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
            <idb_stat_mkpdu_tx_success>2808</idb_stat_mkpdu_tx_success>
            <idb_stat_mkpdu_tx_fail>0</idb_stat_mkpdu_tx_fail>
            <idb_stat_mkpdu_tx_pkt_build_fail>0</idb_stat_mkpdu_tx_pkt_build_fail>
            <idb_stat_mkpdu_no_tx_on_intf_down>0</idb_stat_mkpdu_no_tx_on_intf_down>
            <idb_stat_mkpdu_no_rx_on_intf_down>0</idb_stat_mkpdu_no_rx_on_intf_down>
            <idb_stat_mkpdu_rx_ca_notfound>0</idb_stat_mkpdu_rx_ca_notfound>
            <idb_stat_mkpdu_rx_error>0</idb_stat_mkpdu_rx_error>
            <idb_stat_mkpdu_rx_success>2714</idb_stat_mkpdu_rx_success>
            <idb_stat_mkpdu_failure_rx_integrity_check_error>0</idb_stat_mkpdu_
failure_rx_integrity_check_error>
            <idb_stat_mkpdu_failure_invalid_peer_mn_error>0</idb_stat_mkpdu_fai
lure_invalid_peer_mn_error>
            <idb_stat_mkpdu_failure_nonrecent_peerlist_mn_error>1</idb_stat_mkp
du_failure_nonrecent_peerlist_mn_error>
            <idb_stat_mkpdu_failure_sakuse_kn_mismatch_error>0</idb_stat_mkpdu_
failure_sakuse_kn_mismatch_error>
<idb_stat_mkpdu_failure_sakuse_rx_not_set_error>0</idb_stat_mkpdu_f
```

```
ailure_sakuse_rx_not_set_error>
                <idb_stat_mkpdu_failure_sakuse_key_mi_mismatch_error>0</idb_stat_mk
pdu_failure_sakuse_key_mi_mismatch_error>
                <idb_stat_mkpdu_failure_sakuse_an_not_in_use_error>0</idb_stat_mkpd
u_failure_sakuse_an_not_in_use_error>
                <idb_stat_mkpdu_failure_sakuse_ks_rx_tx_not_set_error>0</idb_stat_m
kpdu_failure_sakuse_ks_rx_tx_not_set_error>
                <idb_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>0</id
b_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>
                <idb_stat_sak_failure_sak_generate_error>0</idb_stat_sak_failure_sa
k_generate_error>
                <idb_stat_sak_failure_hash_generate_error>0</idb_stat_sak_failure_h
ash_generate_error>
                <idb_stat_sak_failure_sak_encryption_error>0</idb_stat_sak_failure_
sak_encryption_error>
                <idb_stat_sak_failure_sak_decryption_error>0</idb_stat_sak_failure_
sak_decryption_error>
                <idb_stat_sak_failure_ick_derivation_error>0</idb_stat_sak_failure_
ick_derivation_error>
                <idb_stat_sak_failure_kek_derivation_error>0</idb_stat_sak_failure_
kek_derivation_error>
                <idb_stat_sak_failure_invalid_macsec_capability_error>0</idb_stat_s
ak_failure_invalid_macsec_capability_error>
                <idb_stat_macsec_failure_rx_sa_create_error>0</idb_stat_macsec_fail
ure_rx_sa_create_error>
                <idb_stat_macsec_failure_tx_sa_create_error>0</idb_stat_macsec_fail
ure_tx_sa_create_error>
              </ROW_idb_stats>
             </TABLE_idb_stats>
           </ROW_mka_intf_stats>
          </TABLE_mka_intf_stats>
         </__readonly__>
        </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
       </__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
      </statistics>
     </mka>
    </macsec>
   </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 4: Displays the MACsec MKA configuration.

```
switch# show macsec mka summary | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <mka>
     <__XML__OPT_Cmd_some_macsec_summary>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
        <TABLE_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet2/1</ifname>
          <policy>am2</policy>
<keychain>kc2/0200000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet3/1</ifname>
```

```
            <policy>am2</policy>
            <keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
           </ROW_mka_summary>

[TRUNCATED FOR READABILITY]

<ROW_mka_summary>
           <ifname>Ethernet3/32</ifname>
           <policy>am2</policy>
           <keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
           </ROW_mka_summary>
          </TABLE_mka_summary>
         </__readonly__>
        </__XML__OPT_Cmd_some_macsec___readonly__>
       </__XML__OPT_Cmd_some_macsec_summary>
     </mka>
    </macsec>
   </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 5: Displays the configuration for a specific MACsec policy.

```
switch# show macsec policy am2 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <policy>
     <__XML__OPT_Cmd_some_macsec_policy_name>
      <policy_name>am2</policy_name>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
        <TABLE_macsec_policy>
         <ROW_macsec_policy>
          <name>am2</name>
          <cipher_suite>GCM-AES-XPN-256</cipher_suite>
          <keyserver_priority>0</keyserver_priority>
          <window_size>512</window_size>
          <conf_offset>0</conf_offset>
          <security_policy>must-secure</security_policy>
          <sak-expiry-time>60</sak-expiry-time>
         </ROW_macsec_policy>
        </TABLE_macsec_policy>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec___readonly__>
     </__XML__OPT_Cmd_some_macsec_policy_name>
    </policy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 6: Displays MACsec security statistics.

```
switch# show macsec secy statistics interface ethernet 4/31 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <secy>
     <statistics>
      <interface>
       <__XML__INTF_ifname>
        <__XML__PARAM_value>
         <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
        </__XML__PARAM_value>
        <__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
         <__readonly__>
          <TABLE_statistics>
           <ROW_statistics>
            <in_pkts_unicast_uncontrolled>0</in_pkts_unicast_uncontrolled>
            <in_pkts_multicast_uncontrolled>42</in_pkts_multicast_uncontrolled>
            <in_pkts_broadcast_uncontrolled>0</in_pkts_broadcast_uncontrolled>
            <in_rx_drop_pkts_uncontrolled>0</in_rx_drop_pkts_uncontrolled>
            <in_rx_err_pkts_uncontrolled>0</in_rx_err_pkts_uncontrolled>
            <in_pkts_unicast_controlled>0</in_pkts_unicast_controlled>
            <in_pkts_multicast_controlled>2</in_pkts_multicast_controlled>
            <in_pkts_broadcast_controlled>0</in_pkts_broadcast_controlled>
            <in_rx_drop_pkts_controlled>0</in_rx_drop_pkts_controlled>
            <in_rx_err_pkts_controlled>0</in_rx_err_pkts_controlled>
            <in_octets_uncontrolled>7230</in_octets_uncontrolled>
            <in_octets_controlled>470</in_octets_controlled>
            <input_rate_uncontrolled_pps>0</input_rate_uncontrolled_pps>
            <input_rate_uncontrolled_bps>9</input_rate_uncontrolled_bps>
            <input_rate_controlled_pps>0</input_rate_controlled_pps>
            <input_rate_controlled_bps>23</input_rate_controlled_bps>
            <out_pkts_unicast_uncontrolled>0</out_pkts_unicast_uncontrolled>
            <out_pkts_multicast_uncontrolled>41</out_pkts_multicast_uncontrolled>
            <out_pkts_broadcast_uncontrolled>0</out_pkts_broadcast_uncontrolled>
            <out_rx_drop_pkts_uncontrolled>0</out_rx_drop_pkts_uncontrolled>
            <out_rx_err_pkts_uncontrolled>0</out_rx_err_pkts_uncontrolled>
            <out_pkts_unicast_controlled>0</out_pkts_unicast_controlled>
            <out_pkts_multicast_controlled>2</out_pkts_multicast_controlled>
            <out_pkts_broadcast_controlled>0</out_pkts_broadcast_controlled>
            <out_rx_drop_pkts_controlled>0</out_rx_drop_pkts_controlled>
            <out_rx_err_pkts_controlled>0</out_rx_err_pkts_controlled>
            <out_octets_uncontrolled>6806</out_octets_uncontrolled>
            <out_octets_controlled>470</out_octets_controlled>
            <out_octets_common>7340</out_octets_common>
            <output_rate_uncontrolled_pps>2598190092</output_rate_uncontrolled_pps>
            <output_rate_uncontrolled_bps>2598190076</output_rate_uncontrolled_bps>
            <output_rate_controlled_pps>0</output_rate_controlled_pps>
            <output_rate_controlled_bps>23</output_rate_controlled_bps>
            <in_pkts_transform_error>0</in_pkts_transform_error>
            <in_pkts_control>40</in_pkts_control>
            <in_pkts_untagged>0</in_pkts_untagged>
            <in_pkts_no_tag>0</in_pkts_no_tag>
            <in_pkts_badtag>0</in_pkts_badtag>
            <in_pkts_no_sci>0</in_pkts_no_sci>
            <in_pkts_unknown_sci>0</in_pkts_unknown_sci>
            <in_pkts_tagged_ctrl>0</in_pkts_tagged_ctrl>
            <out_pkts_transform_error>0</out_pkts_transform_error>
            <out_pkts_control>41</out_pkts_control>
            <out_pkts_untagged>0</out_pkts_untagged>
            <rx_sa_an>1</rx_sa_an>
            <in_pkts_unchecked>0</in_pkts_unchecked>
            <in_pkts_delayed>0</in_pkts_delayed>
            <in_pkts_late>0</in_pkts_late>
```

```
                    <in_pkts_ok>1</in_pkts_ok>
                    <in_pkts_invalid>0</in_pkts_invalid>
                    <in_pkts_not_valid>0</in_pkts_not_valid>
                    <in_pkts_not_using_sa>0</in_pkts_not_using_sa>
                    <in_pkts_unused_sa>0</in_pkts_unused_sa>
                    <in_octets_decrypted>223</in_octets_decrypted>
                    <in_octets_validated>0</in_octets_validated>
                    <tx_sa_an>1</tx_sa_an>
                    <out_pkts_encrypted_protected>1</out_pkts_encrypted_protected>
                    <out_pkts_too_long>0</out_pkts_too_long>
                    <out_pkts_sa_not_inuse>0</out_pkts_sa_not_inuse>
                    <out_octets_encrypted_protected>223</out_octets_encrypted_protected>
                  </ROW_statistics>
                </TABLE_statistics>
              </__readonly__>
            </__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
          </__XML__INTF_ifname>
        </interface>
      </statistics>
    </secy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 7: Displays the running configuration information for MACsec.

```
switch# show running-config macsec | xml

!Command: show running-config macsec
!Time: Fri Jan 20 07:12:34 2017

version 7.0(3)I4(6)
*******************************************
This may take time. Please be patient.
*******************************************
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cis
co.com/nxos:7.0.3.I4.6.:configure_" xmlns:m="http://www.cisco.com/nxos:7.0.3.I4.
6.:_exec" xmlns:m1="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__macsec-poli
cy" xmlns:m2="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__if-eth-non-member
" message-id="1">
  <nf:get-config>
    <nf:source>
     <nf:running/>
    </nf:source>
    <nf:filter>
      <m:configure>
        <m:terminal>
          <feature>
            <macsec/>
          </feature>
          <macsec>
            <policy>
              <__XML__PARAM__policy_name>
                <__XML__value>am2</__XML__value>
                <m1:cipher-suite>
                  <m1:__XML__PARAM__suite>
                    <m1:__XML__value>GCM-AES-XPN-256</m1:__XML__value>
                  </m1:__XML__PARAM__suite>
                </m1:cipher-suite>
                <m1:key-server-priority>
                  <m1:__XML__PARAM__pri>
```

```
                                    <m1:__XML__value>0</m1:__XML__value>
                                  </m1:__XML__PARAM__pri>
                              </m1:key-server-priority>
              <m1:window-size>
              <m1:__XML__PARAM__size>
                                    <m1:__XML__value>512</m1:__XML__value>
                                  </m1:__XML__PARAM__size>
                              </m1:window-size>
                              <m1:conf-offset>
                                <m1:__XML__PARAM__offset>
                                  <m1:__XML__value>CONF-OFFSET-0</m1:__XML__value>
                                </m1:__XML__PARAM__offset>
                              </m1:conf-offset>
                              <m1:security-policy>
                                <m1:__XML__PARAM__policy>
                                  <m1:__XML__value>must-secure</m1:__XML__value>
                                </m1:__XML__PARAM__policy>
                              </m1:security-policy>
                              <m1:sak-expiry-time>
                                <m1:__XML__PARAM__ts>
                                  <m1:__XML__value>60</m1:__XML__value>
                                </m1:__XML__PARAM__ts>
                              </m1:sak-expiry-time>
                            </__XML__PARAM__policy_name>
                        </policy>
                    </macsec>
                    <interface>
                      <__XML__PARAM__interface>
                        <__XML__value>Ethernet2/1</__XML__value>
                        <m2:macsec>
                          <m2:keychain>
                            <m2:__XML__PARAM__keychain_name>
                              <m2:__XML__value>kc2</m2:__XML__value>
                              <m2:policy>
                                <m2:__XML__PARAM__policy_name>
                                  <m2:__XML__value>am2</m2:__XML__value>
                                </m2:__XML__PARAM__policy_name>
                              </m2:policy>
                            </m2:__XML__PARAM__keychain_name>
                          </m2:keychain>
                        </m2:macsec>
                      </__XML__PARAM__interface>
                    </interface>

[TRUNCATED FOR READABILITY]

<interface>
            <__XML__PARAM__interface>
              <__XML__value>Ethernet4/31</__XML__value>
              <m2:macsec>
                <m2:keychain>
                  <m2:__XML__PARAM__keychain_name>
                    <m2:__XML__value>kc2</m2:__XML__value>
                    <m2:policy>
                      <m2:__XML__PARAM__policy_name>
                        <m2:__XML__value>am2</m2:__XML__value>
                      </m2:__XML__PARAM__policy_name>
                    </m2:policy>
                  </m2:__XML__PARAM__keychain_name>
                </m2:keychain>
              </m2:macsec>
            </__XML__PARAM__interface>
          </interface>
        </m:terminal>
```

```
        </m:configure>
      </nf:filter>
    </nf:get-config>
  </nf:rpc>
]]>]]>
```

# MIBs

MACsec supports the following MIBs:

- IEEE8021-SECY-MIB

- CISCO-SECY-EXT-MIB

To locate and download supported MIBs, go to the following URL:
ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html.

# Related Documentation

| Related Topic | Document Title |
|---|---|
| Keychain management | Cisco Nexus 9000 Series NX-OS Security Configuration Guide |
| System messages | Cisco Nexus 9000 Series NX-OS System Messages References |

# Configuring TCP Authentication Option

This document describes how to configure TCP authentication option on Cisco NX-OS devices.

## About TCP Authentication Option

With TCP Authentication Option (TCP-AO), defined in RFC 5925, you can protect long-lived TCP connections against replays using stronger Message Authentication Codes (MACs).

TCP-AO is the proposed replacement for TCP MD5, defined in RFC 2385. Unlike TCP MD5, TCP-AO is resistant to collision attacks and provides algorithmic agility and support for key management.

TCP-AO has the following distinct features:

- TCP-AO supports the use of stronger Message Authentication Codes (MACs) to enhance the security of long-lived TCP connections.

- TCP-AO protects against replays for long-lived TCP connections, and coordinates key changes between endpoints by providing a more explicit key management.

TCP-AO deprecates TCP MD5 however to support legacy TCP peers, NXOS and BGP will continue to support TCP-MD5 for legacy peers. However, a configuration in which one of the devices is configured with the TCP MD5 option and the other with the TCP-AO option is not supported.

## TCP-AO Key Chain

TCP-AO is based on traffic keys and Message Authentication Codes (MACs) generated using the keys and a MAC algorithm. The traffic keys are derived from master keys that you can configure in a TCP-AO key chain. Use the **key chain** *key-chain-name* **tcp** command in the global configuration mode to create a TCP-AO

key chain and configure keys in the chain. The TCP-AO key chain must be configured on both the peers communicating via a TCP connection.

Keys in a TCP-AO key chain have the following configurable properties:

| Configurable Property | Description |
|---|---|
| send-id | Key identifier of the TCP-AO option of the outgoing segment. |
| | The send identifier configured on a router must match the receive identifier configured on the peer. |
| recv-id | Key identifier compared with the TCP-AO key identifier of the incoming segment during authentication. |
| | The receive identifier configured on a router must match the send identifier configured on the peer. |
| cryptographic-algorithm | The MAC algorithm to be used to create MACs for outgoing segments. The algorithm can be one of the following: |
| | • AES-128-CMAC authentication algorithm |
| | • HMAC-SHA-1 authentication algorithm |
| | • HMAC-SHA-256 authentication algorithm. |
| include-tcp-options | This flag indicates whether TCP options other than TCP-AO will be used to calculate MACs. |
| | With this flag enabled, the contents of all options along with a zero-filled authentication option, is used to calculate the MAC. |
| | When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations. |
| | This flag is disabled by default. |
| | **Note** The configuration of this flag is overridden by the application configuration when the application configuration is available. |
| send-lifetime | This configuration determines the time for which a key is valid and can be used for TCP-AO-based authentication of TCP segments. When the lifetime of key elapses and the key expires, the next key with the youngest lifetime is selected. |
| key-string | The key string is a pre-shared master key configured on both peers and is used to derive the traffic keys. |

**TCP-AO Format**

```
+------------+------------+------------+------------+
| Kind=29    |   Length   |   KeyID    | RNextKeyID |
+------------+------------+------------+------------+
|                     MAC           ...
+--------------------------------...

   ...----------------+
```

```
...   MAC (con't)    |
...-----------------+
```

The fields of the TLV format are as follows:

- Kind: Indicates TCP-AO with a value of 29.

- Length: Indicates the length of the TCP-AO sequence.

- KeyID: The send identifier of the MKT that was used to generate the traffic keys.

- RNextKeyID: The receive identifier of the MKT that is ready to be used to authenticate received segments.

- MAC: The MAC computed for the TCP segment data and the prefixed pseudo header.

### Master Key Tuples

Traffic keys are the keying material used to compute the message authentication codes of individual TCP segments.

Master Key Tuples (MKTs) enable you to derive unique traffic keys, and to include the keying material required to generate those traffic keys. MKTs indicate the parameters under which the traffic keys are configured. The parameters include whether TCP options are authenticated, and indicators of the algorithms used for traffic key derivation and MAC calculation.

Each MKT has two identifiers, namely **SendID** and a **RecvID** . The SendID identifier is inserted as the KeyID identifier of the TCP AO option of the outgoing segments.The **RecvID** is matched against the TCP AO KeyID of the incoming segments.

# TCP-AO Key Rollover

TCP-AO keys are valid for a defined duration configured using the send-lifetime. If send-lifetime is not configured the key is considered inactive. Key rollover is initiated based on the send lifetimes of keys.

TCP-AO coordinates use of new MKTs using the RNextKeyID and KeyID field on the TCP-AO option field. For hitless key rollovers, new and old keys in keychain configurations need to have at least 15 minutes of overlap. This is required so that the TCP-AO has enough time to coordinates use of new MKT.

When key rollover is initiated, one of the peer routers, say Router A, indicates that the rollover is necessary. To indicate that the rollover is necessary, Router A sets the RNextKeyID to the receive identifier (recv-id) of the new MKT to be used. On receiving the TCP segment, the peer router, say Router B, looks up the send identifier (send-id) in its database to find the MKT indicated by the RNextKeyID in the TCP-AO payload. If the key is available and valid, Router B sets the current key to the new MKT. After Router B has rolled over, Router A also sets the current key to the new Primary Key Tuples.

Key rollover is initiated with overlapping send-lifetimes and send-lifetime expiry

If you do not configure a new key that can be activated before the expiry of the current key, the key may time out and expire. Such an expiry can cause retransmissions with the peer router rejecting segments authenticated with the expired key. The connection may fail due to Retransmission Time Out (RTO). When new valid keys are configured, a new connection is established.

# Guidelines and Limitations

- The send-id and recv-id of each key in the key chain must be unique. Because send-id and recv-id must be chosen from the range 0 to 255, the TCP-AO key chain can have a maximum of 256 keys.

- Only one keychain can be associated with an application connection. Rollover is always performed within the keys in this keychain.

- If the key in use expires, expect segment loss until a new key that has a valid lifetime is configured on each side and keys rollover.

- All the following configurations must be done for a TCP-AO keychain key to be considered active: send-id, recv-id, key-string, send-lifetime and cryptographic-algorithm.

- Keychain infra picks up youngest key based on send-lifetime configuration. Or whichever key was configured last if same send-lifetime is configured for two keys. Ideally, we should not do that.

- User MUST configure minimum 15 minutes overlapping time between the two overlapping keys.

- Modifying the configuration of a key in use such as key-string, send-id, recv-id, cryptographic-algorithm or send-lifetime will result in TCP connection flap.

- A keychain's configuration type must match the type it has been linked to within the client protocol. If an attempt is made to mismatch these types, a syslog message is generated to notify the user. For example: It is not supported if a keychain named keychain_abc is configured as a Macsec keychain but is associated as a TCP keychain with BGP. Similarly, the case where the keychain is first associated with the client (a process known as forward-referencing) and then configured as a different keychain type, is also not supported.

# Configure TCP Key Chain and Keys

**Before you begin**

- Ensure that the key-string, send-lifetimes, cryptographic-algorithm, and ids of keys match on both peers.

- Ensure that the send-id on a router matches the recv-id on the peer router. We recommend using the same id for both the parameters unless there is a need to use separate key spaces.

- The send-id and recv-id of a key cannot be reused for another key in the same key chain.

- The key-string is encrypted and stored in Type-6 format if AES password encryption feature is enabled and primary key configured otherwise it will be stored in Type-7 encrypted format.

- For more details, see Configuring a Primary Key and Enabling the AES Password Encryption Feature, on page 517

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br><br>**Example:**<br><br>switch# configure terminal | Enters global configuration mode. |
| **Step 2** | **key chain** *name* **tcp**<br><br>**Example:**<br><br>switch(config)# key chain bgp-keys tcp | Enters keychain configuration mode for the keychain that you specified. |
| **Step 3** | **key** *key-ID*<br><br>**Example:**<br><br>switch(config-tcpkeychain)# key 13 | Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535. |
| **Step 4** | **send-id** *send-ID*<br><br>**Example:**<br><br>switch(config-tcpkeychain-tcpkey)# send-id 2 | Specifies the send identifier for the key. The send-ID must be in the range from 0 to 255 and unique value per key chain. |
| **Step 5** | **recv-id** *recv-ID*<br><br>**Example:**<br><br>switch(config-tcpkeychain-tcpkey)# recv-id 2 | Specifies the recieve identifier for the key. The recv-ID must be in the range from 0 to 255 and unique value per key chain. |
| **Step 6** | **key-string** *[encryption-type] text-string*<br><br>**Example:**<br><br>switch(config-tcpkeychain-tcpkey)# key-string 0 AS3cureStr1ng | Configures the text string for the key. The text-string argument is alphanumeric, case-sensitive, and supports special characters.<br><br>The encryption-type argument can be one of the following values:<br><br>• 0—The text-string argument that you enter is unencrypted text. This is the default.<br><br>• 6—Beginning with Cisco NX-OS Release 10.3(3)F, the Cisco proprietary (Type-6 encrypted) method is supported on Cisco Nexus 9000 Series platform switches.<br><br>• 7—The text-string argument that you enter is encrypted. The encryption method is a Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Cisco NX-OS device.<br><br>The **key-string** command has limitations on using the following special characters in the *text-string*: |

| Command or Action | Purpose |
|---|---|
| | <table><tr><td>Special Character</td><td>Description</td><td>Comments</td></tr><tr><td>\|</td><td>Vertical bar or pipe</td><td>Unsupported at start of key-string</td></tr><tr><td>></td><td>Greater than</td><td>Unsupported at start of key-string</td></tr><tr><td>\</td><td>Backslash</td><td>Unsupported start or end of a key-string</td></tr><tr><td>(</td><td>Left parenthesis</td><td>Unsupported at start of key-string</td></tr><tr><td>'</td><td>Apostrophe</td><td>Unsupported at start of key-string</td></tr><tr><td>"</td><td>Quotation mark</td><td>Unsupported at start of key-string</td></tr><tr><td>?</td><td>Question mark</td><td>Supported. However, press **Ctrl-V** before entering a question mark (?).</td></tr></table> For more information on the special characters usage in commands, see Understanding the Command-Line Interface section. |
| **Step 7** | **[no] cryptographic-algorithm {HMAC-SHA-1 \| HMAC-SHA-256 \| AES-128-CMAC }**<br><br>**Example:**<br>`switch(config-tcpkeychain-tcpkey)#`<br>`cryptographic-algorithm HMAC-SHA-1` | Specifies the algorithm to be used to compute MACs for TCP segments. You can configure only one cryptographic algorithm per key. |
| **Step 8** | **send-lifetime [local]** *start-time* **duration** [duration-value \| infinite \| end-time]<br><br>**Example:**<br>`switch(config-tcpkeychain-tcpkey)#`<br>`send-lifetime local 01:01:01 Jan 01 2023`<br>`01:01:01 Jan 10 2023` | Configures a send lifetime for the key. By default, the device treats the start-time and end-time arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.<br><br>The start-time argument is the time of day and date that the key becomes active.<br><br>You can specify the end of the send lifetime with one of the following options:<br><br>• **duration** duration-value —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years). |

| | Command or Action | Purpose |
|---|---|---|
| | | • **infinite**—The send lifetime of the key never expires.<br><br>• end-time —The end-time argument is the time of day and date that the key becomes inactive. |
| **Step 9** | (Optional) **include-tcp-options**<br><br>**Example:**<br><br>switch(config-tcpkeychain-tcpkey)# include-tcp-options | An option for user to specify if the 'TCP option headers (other than TCP AO option) needs to be included while computing the 'MAC' digest of the packets. |

# Verifying the TCP Keychain

| Command | Purpose |
|---|---|
| **show key chain** *[name] [detail]* | Displays the keychains configured on the device. |

```
switch# show key chain
Key-Chain bgp_keys tcp
  Key 2 -- text 7 "070e234f"
    send-id 2
    recv-id 2
    cryptographic-algorithm AES_128_CMAC
    send lifetime UTC (08:17:00 May 29 2023)-(08:21:00 May 29 2023)
    include-tcp-options
  Key 3 -- text 7 "070c2058"
    send-id 3
    recv-id 4
    cryptographic-algorithm HMAC-SHA-1
    send lifetime UTC (08:20:00 May 29 2023)-(always valid) [active]
    include-tcp-options
  Key 12 -- text ""
    send lifetime UTC (08:20:00 May 29 2023)-(always valid)
```

> ✎
>
> **Note**  [active] indicates that the key is valid and active otherwise the key is inactive. In the above example only key 3 is active and usable.

The show key chain detail command will explicitly display inactive key. In case of type6 encryption the show key chain detail command will display if the type6 key-string is decryptable or not. It will also display `youngest active send key` that client is currently using to authenticate its packets.

```
switch# show key chain detail
Key-Chain bgp_keys tcp
  Key 1 -- text 6 "JDYk9k4kmaciqaH6Eu2+9C0tmCRl9k7JAMYs/fXGbW1lmHP88PAA=="
    Type6 Decryptable: yes
    send-id 1
    recv-id 1
    cryptographic-algorithm HMAC-SHA-1
    send lifetime local (18:15:42 May 15 2023)-(always valid) [active]
    include-tcp-options
    accept-ao-mismatch
```

```
        Key 2 -- text 6 "JDYkB+Fs8u3ujRDpFSu4tH6H7iTS45JJA6sKeGsBD0L3HjGDeg9AA=="
          Type6 Decryptable: yes
          send-id 2
          recv-id 2
          cryptographic-algorithm AES_128_CMAC
          send lifetime local (17:10:47 May 15 2023)-(18:15:42 May 15 2023) [inactive]

      youngest active send key: 1
```

# Configuration Example for TCP Keychain

This example shows how to configure a TCP keychain named bgp_keys. Each key text string is encrypted. The keys have overlapping lifetime configurations:

```
key chain bgp_keys tcp
  key 1
    send-id 1
    recv-id 1
    key-string 7 070e234f
    send-lifetime 01:00:00 Oct 10 2023 01:00:00 Oct 11 2023
    cryptographic-algorithm AES-128-CMAC
  key 2
    send-id 2
    recv-id 2
    key-string 7 075e731f
    send-lifetime 00:45:00 Oct 11 2023 01:00:00 Oct 12 2023
    cryptographic-algorithm HMAC-SHA-256
    include-tcp-options
```

# I N D E X