Inspur

CN93240YC-FX2

NX-OS Security Configuration Guide

(Release 9.3.x)

Inspur-Cisco Networking Technology Co.,Ltd. provides customers with comprehensive technical support and services. For any assistance, please contact our local office or company headquarters.
Website: http://www.inspur.com/
Technical Support Tel: 400-691-1766
Technical Support Email:icnt_service@inspur.com
Technical Document Support Email: icnt_service@inspur.com
Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province
Postal code: 250101

Preface

Objectives

This guide describes main functions of the CN93240YC-FX2. To have a quick grasp of the CN93240YC-FX2, please read this manual carefully.

Versions

The following table lists the product versions related to this document.

Product name	Version
CN93240YC-FX2	

Conventions

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
⚠ Warning	Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury.
⚠ Caution	Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results.
✎ Note	Provides additional information to emphasize or supplement important points of the main text.
🔍 Tip	Indicates a tip that may help you solve a problem or save time.

General conventions

Convention	Description
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.
Italic	Book titles are in *italics*.
`Lucida Console`	Terminal display is in `Lucida Console`.

Command conventions

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x	y	... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x	y	...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x	y	... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x	y	...] *	The parameter before the & sign can be repeated 1 to n times.

GUI conventions

Convention	Description
Boldface	Buttons, menus, parameters, tabs, windows, and dialog titles are in **boldface**. For example, click **OK**.
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose **File** > **Create** > **Folder**.

Keyboard operation

Format	Description
Key	Press the key. For example, press **Enter** and press **Tab**.

Format	Description
Key 1+Key 2	Press the keys concurrently. For example, pressing **Ctrl+C** means the two keys should be pressed concurrently.
Key 1, Key 2	Press the keys in turn. For example, pressing **Alt**, **A** means the two keys should be pressed in turn.

Mouse operation

Action	Description
Click	Select and release the primary mouse button without moving the pointer.
Double-click	Press the primary mouse button twice continuously and quickly without moving the pointer.
Drag	Press and hold the primary mouse button and move the pointer to a certain position.

Change history

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Issue 01 (2020-02-24)

Initial commercial release

CONTENTS

CHAPTER 4 **Configuring RADIUS** **39**

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain CN 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.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to icnt_service@inspur.com. We appreciate your feedback.

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *CN93240YC-FX2 NX-OS Security Guide, Release 9.3(x)*.

New and Changed Information

This table summarizes the new and changed features for the *CN93240YC-FX2 NX-OS Security Configuration Guide, Release 9.3(x)* and where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
DHCP	Added the ability for the DHCP relay to choose either the primary or the secondary subnet when the interface includes both these subnets.	9.3(3)	Configuring DHCP Relay Subnet-Selection
DHCPv6	Added support for DHCPv6 Option 79 (client link-layer address) in DHCPv6-relayed packets for CN93240YC-FX2 switches.	9.3(3)	Enabling DHCPv6 Option 79 Additional References for DHCP
User-defined MAC address	Enables you to configure a user-defined MAC address limit between the range of 16 to 256.	9.3(2)	Guidelines and Limitations for MAC ACLs Configuring an Interface MAC Address and Limit
DHCP	Added the ability to disable the server identifier override option for DHCP Option 82 packets.	9.3(3)	Disabling the Server Identifier Override Option

Feature	Description	Changed in Release	Where Documented
MAC UDF	Ability to enables the device to match on user-defined fields (UDFs) and to apply the matching packets to MAC ACLs.	9.3(2)	Configuring a UDF-Based MAC ACL

Inspur-Cisco Networking
Technology Co.,Ltd.

Overview

The 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:

- Authentication, Authorization, and Accounting
- RADIUS and TACACS+ Security Protocols
- LDAP
- SSH and Telnet
- User Accounts and Roles
- IP ACLs
- MAC ACLs
- VACLs
- DHCP Snooping
- Dynamic ARP Inspection
- IP Source Guard
- Password Encryption
- Keychain Management
- Control Plane Policing
- Rate Limits
- Software Image
- Virtual Device Contexts=

Authentication, Authorization, and Accounting

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

Authentication

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

Authorization

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

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

Accounting

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

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

Related Topics

Configuring AAA

RADIUS and TACACS+ Security Protocols

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

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

RADIUS

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

TACACS+

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

Related Topics

Configuring RADIUS

Configuring TACACS+

LDAP

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

Related Topics

Configuring LDAP

SSH and Telnet

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

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

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

Related Topics

Configuring SSH and Telnet

User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the 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.

Related Topics

Configuring User Accounts and RBAC

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 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 NX-OS software applies the applicable default rule. The NX-OS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Configuring IP ACLs

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 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 NX-OS software applies the applicable default rule. The NX-OS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Configuring MAC ACLs

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).

Related Topics

Configuring VLAN ACLs

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 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 master 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.

Related Topics

Configuring Password Encryption

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.

Related Topics

Configuring Keychain Management

Control Plane Policing

The NX-OS device provides control plane policing to prevent denial-of-service (DoS) attacks from impacting performance. The supervisor module of the 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 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.

Related Topics

Configuring Control Plane Policing

Rate Limits

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

Related Topics

Configuring Rate Limits

Virtual Device Contexts

NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The CN93240YC-FX2 switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

CHAPTER 3

Configuring AAA

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

This chapter includes the following sections:

- About AAA
- Licensing Requirements for AAA
- Prerequisites for AAA
- Guidelines and Limitations for AAA
- Default Settings for AAA
- Configuring AAA
- Monitoring and Clearing the Local AAA Accounting Log
- Verifying the AAA Configuration
- Configuration Examples for AAA
- Configuration Examples for Login Parameters
- Configuration Examples for the Password Prompt Feature
- Additional References for AAA=

About AAA

This section includes information about AAA on 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 NX-OS device. 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, 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 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 NX-OS device, which is based on the user ID and password combination provided by the entity trying to access the NX-OS device. 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 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 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 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 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 NX-OS devices in the fabric.

- It is easier to manage user attributes for each NX-OS device in the fabric than using the local databases on the 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 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 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 2: AAA Service Configuration Commands

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

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

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 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 NX-OS device.

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

Table 3: AAA Authentication Methods for AAA Services

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

Note For console login authentication, user login authentication, and user management session accounting, the 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.
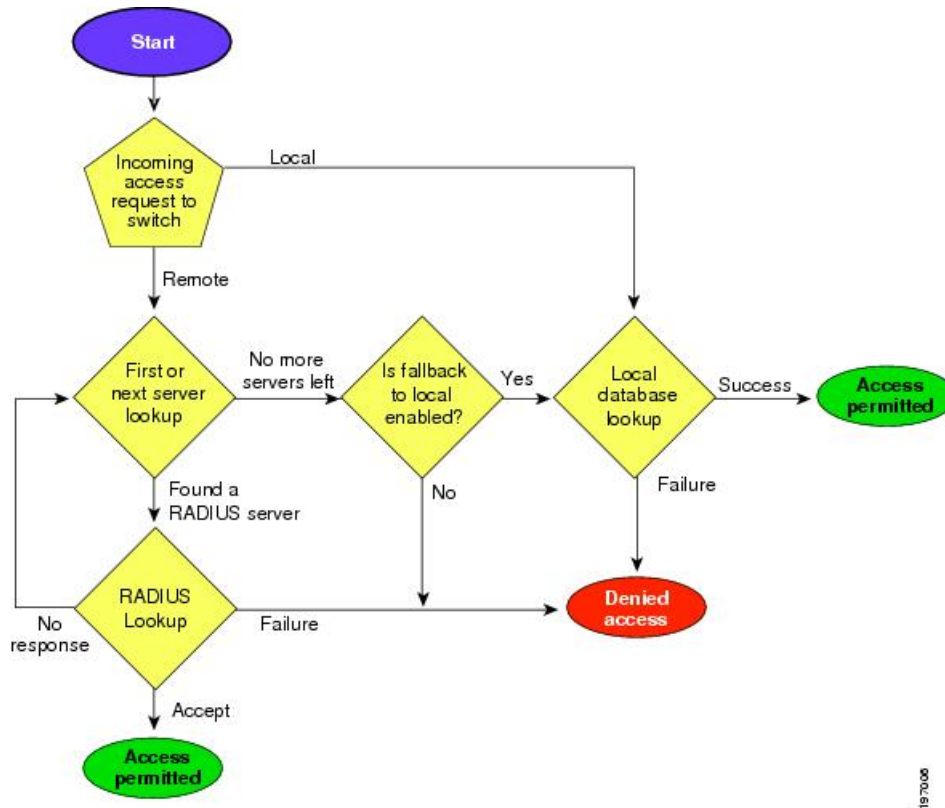
The following list explains the process:

- When you log in to the required 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 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 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 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 Master 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 master encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a master 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 NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Licensing Requirements for AAA

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	AAA 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 NX-OS licensing scheme, see the NX-OS Licensing Guide.

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

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

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

Guidelines and Limitations for AAA

AAA has the following guidelines and limitations:

- If you have a user account configured on the local NX-OS device that has the same name as a remote user account on an AAA server, the 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.

- CN93240YC-FX2 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.=

Default Settings for AAA

This table lists the default settings for AAA parameters.

Table 4: 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 NX-OS devices.

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

Note   CN93240YC-FX2 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 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 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 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 *CN93240YC-FX2 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** **Example:** switch# **configure terminal** switch(config)#	Enters configuration mode.

Step 2	**aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}
	
	Example:
	
	switch(config)# **aaa authentication login console group radius**
		Configures login authentication methods for the console.
		
		The *group-list* argument consists of a space-delimited list of group names. The group names are the following:
		
		radius
		    Uses the global pool of RADIUS servers for authentication.
		named-group
		    Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.
		
		The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used.
		
		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** **Example:** switch(config)# **exit** switch#	Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show aaa authentication** **Example:** switch# **show aaa authentication**	Displays the configuration of the console login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** 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 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** **Example:** switch# **configure terminal** switch(config)#	Enters configuration mode.
Step 2	**aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**} **Example:** switch(config)# **aaa authentication login default group radius**	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: - **radius**—Uses the global pool of RADIUS servers for authentication. - *named-group*—Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.

	Command or Action	Purpose
		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.
		You can configure one of the following:
		• AAA authentication groups
		• AAA authentication groups with no authentication
		• Local authentication
		• No authentication
		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.
Step 3	**exit** **Example:** `switch(config)# `**`exit`** `switch#`	Exits configuration mode.
Step 4	(Optional) **show aaa authentication** **Example:** `switch# `**`show aaa authentication`**	Displays the configuration of the default login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

Command or Action	Purpose
switch# **copy running-config startup-config**	

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 NX-OS device falls back to local authentication to ensure that users are not 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 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** **Example:** switch# **configure terminal** switch(config)#	Enters configuration mode.
Step 2	**no aaa authentication login** {**console** | **default**} **fallback error local** **Example:** switch(config)# **no aaa authentication 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. The following message appears when you disable fallback to local authentication: "WARNING!!! Disabling fallback can lock your switch."
Step 3	(Optional) **exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 4	(Optional) **show aaa authentication** **Example:** switch# **show aaa authentication**	Displays the configuration of the console and default login authentication methods.

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

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

Enabling Login Authentication Failure Messages

When you log in, the login is processed by 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa authentication login error-enable**
	
	Example:
	
	`switch(config)# aaa authentication login`
	`error-enable`
		Enables login authentication failure messages. The default is disabled.

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

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the login failure message configuration.

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

Logging Successful and Failed Login Attempts

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

Procedure

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

	Command or Action	Purpose
Step 2	Required: [**no**] **login on-failure log** **Example:** `switch(config)# login on-failure log`	Logs all failed authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the failed login: AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.
Step 3	Required: [**no**] **login on-success log** **Example:** `switch(config)# login on-success log`	Logs all successful authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the successful login: AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message.
Step 4	(Optional) **show login on-failure log** **Example:** `switch(config)# show login on-failure log`	Displays whether the switch is configured to log failed authentication messages to the syslog server.
Step 5	(Optional) **show login on-successful log** **Example:** `switch(config)# show login on-successful log`	Displays whether the switch is configured to log successful authentication messages to the syslog server.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling CHAP Authentication

The 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 NX-OS device through a remote authentication server (RADIUS or TACACS+).

By default, the NX-OS device uses Password Authentication Protocol (PAP) authentication between the 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 CN93240YC-FX2 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 5: 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** **Example:** switch# **configure terminal** switch(config)#	Enters configuration mode.
Step 2	**no aaa authentication login ascii-authentication** **Example:**	Disables ASCII authentication.

	Command or Action	Purpose
	switch(config)# **no aaa authentication login ascii-authentication**	
Step 3	**aaa authentication login chap enable** **Example:** switch(config)# **aaa authentication login chap enable**	Enables CHAP authentication. The default is disabled. **Note** You cannot enable both CHAP and MSCHAP or MSCHAP V2 on your Cisco NX-OS device.
Step 4	(Optional) **exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 5	(Optional) **show aaa authentication login chap** **Example:** switch# **show aaa authentication login chap**	Displays the CHAP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch# **copy running-config 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 NX-OS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a NX-OS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a 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 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 NX-OS device uses Password Authentication Protocol (PAP) authentication between the 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 6: 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**no aaa authentication login ascii-authentication** **Example:** `switch(config)# no aaa authentication login ascii-authentication`	Disables ASCII authentication.
Step 3	**aaa authentication login** {**mschap** | **mschapv2**} **enable** **Example:** `switch(config)# aaa authentication login mschap enable`	Enables MSCHAP or MSCHAP V2 authentication. The default is disabled. **Note**　You cannot enable both MSCHAP and MSCHAP V2 on your NX-OS device.

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

Step 5	(Optional) **show aaa authentication login** {**mschap** | **mschapv2**} **Example:** `switch# show aaa authentication login mschap`	Displays the MSCHAP or MSCHAP V2 configuration.

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

Configuring AAA Accounting Default Methods

NX-OS software supports TACACS+ and RADIUS methods for accounting. 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 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa accounting default** {**group** *group-list* | **local**} **Example:** `switch(config)# aaa accounting default group radius`	Configures the default accounting method. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for accounting.

	Command or Action	Purpose
		• *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is **local**, which is used when no server groups are configured or when all the configured server groups fail to respond.

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

Step 4	(Optional) **show aaa accounting** **Example:** `switch# show aaa accounting`	Displays the configuration AAA accounting default methods.

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

Using AAA Server VSAs with Cisco NX-OS Devices

You can use vendor-specific attributes (VSAs) to specify 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 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 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 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 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 Inspur 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 inspur-av-pair on AAA servers to specify user role mapping for the NX-OS device using this format:

```
shell:roles="roleA roleB …"
```

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

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

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

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

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** **Example:** switch# **configure terminal**	Enters global configuration mode.
Step 2	[**no**] **system login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** switch(config)# **system login block-for 100 attempts 2 within 60**	Configures the quiet mode time period. The range for all arguments is from 1 to 65535. 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. 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. **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* **Example:** 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.

	Command or Action	**Purpose**
Step 4	(Optional) **show system login** [**failures**] **Example:** `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** **Example:** `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** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	[**no**] **user max-logins** *max-logins* **Example:** `switch(config)# user max-logins 1`	Restricts the maximum 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. **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** **Example:** `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** **Example:** `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**	Enters global configuration mode.
	Example:	
	switch# **configure terminal**	
Step 2	[**no**] **userpassphrase** {**min-length** *min-length* | **max-length** *max-length*}	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.
	Example:	
	switch(config)# **userpassphrase min-length 8 max-length 80**	
Step 3	(Optional) **show userpassphrase** {**length** | **max-length** | **min-length**}	Displays the minimum and maximum length of the user password.
	Example:	
	switch(config)# **show userpassphrase length**	
Step 4	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.
	Example:	
	switch(config)# **copy running-config startup-config**	

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**	Enters global configuration mode.
	Example:	
	switch# **configure terminal**	
Step 2	**password prompt username**	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
	Example:	
	switch(config)# **password prompt username** Password prompt username is enabled. After providing the required options in	

	Command or Action	Purpose
	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.	the **no** form of this command to disable this feature.
Step 3	(Optional) **copy running-config startup-config** **Example:** switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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** **Example:** switch# **configure terminal**	Enters global configuration mode.
Step 2	**generate type7_encrypted_secret** **Example:** switch(config)# **generate type7_encrypted_secret** Type-7 (Vigenere) Encryption, Use this encrypted secret to configure radius and tacacs shared secret with key type 7. Copy complete secret with double quotes. Enter plain text secret: Confirm plain text secret: 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. **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** **Example:** switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Monitoring and Clearing the Local AAA Accounting Log

The 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*] **Example:** `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**] **Example:** `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. **Note** The **clear login failures** command clears the login failures in the current watch period.

Command	Purpose
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.

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
NX-OS Licensing	*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.	—

C H A P T E R **4**

Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on 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 implementation, RADIUS clients run on 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 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 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 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 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 NX-OS device marks unresponsive RADIUS servers as dead

and does not send AAA requests to any dead RADIUS servers. The 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 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.
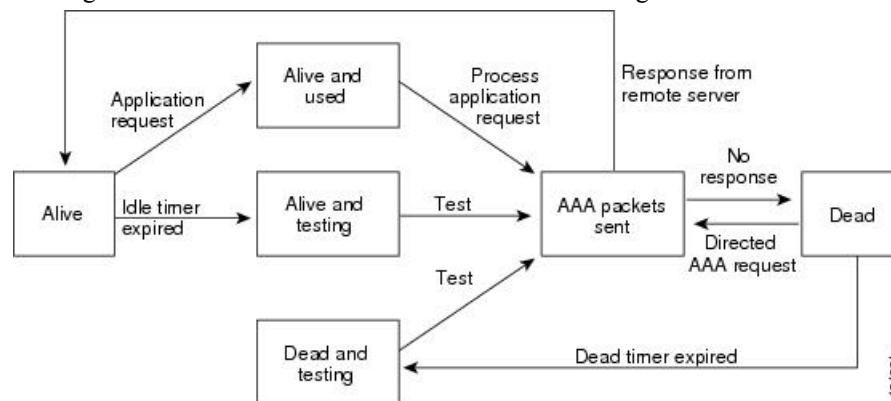
Note   The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

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 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 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 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 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 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 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. 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 VSA and one or more session identification attributes. The VSA is in the form of Inspur: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.

Licensing Requirements for RADIUS

This table shows the licensing requirements for this feature.

Product	License Requirement
NX-OS	RADIUS 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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 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 NX-OS device.

• If you have a user account configured on the local NX-OS device that has the same name as a remote user account on an AAA server, the 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.

• CN93240YC-FX2 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.=

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 7: 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

Parameters	Default
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 NX-OS device.

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

Note CN93240YC-FX2switches 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 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
Configuring Global RADIUS Keys

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

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** 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.
Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** switch(config)# **show radius pending**	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** switch(config)# **radius commit**	Applies the RADIUS configuration changes in the temporary database to the running configuration.
Step 5	**exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 6	(Optional) **show radius-server** **Example:**	Displays the RADIUS server configuration.

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

Related Topics

Configuring a Key for a Specific RADIUS Server

Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the NX-OS device. A RADIUS key is a shared secret text string between the 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** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**radius-server key** [**0** | **6** | **7**] *key-value* **Example:** switch(config)# **radius-server key 0 QsEfThUkO** **Example:** switch(config)# **radius-server key 7 "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 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. **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+.

	Command or Action	**Purpose**

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

About AES Password Encryption and Master Encryption Keys

Configuring a Key for a Specific RADIUS Server

You can configure a key on the NX-OS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** [**0** | **6** | **7**] *key-value*	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 NX-OS

	Command or Action	Purpose
	Example: switch(config)# **radius-server host 10.10.1.1 key 0 PlIjUhYg** **Example:** switch(config)# **radius-server host 10.10.1.1 key 7 "fewhg"**	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. This RADIUS key is used instead of the global RADIUS 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+.
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 Hosts

About AES Password Encryption and Master Encryption Keys

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** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**aaa group server radius** *group-name* **Example:** switch(config)# **aaa group server radius RadServer** 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.
Step 3	**server** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** switch(config-radius)# **server 10.10.1.1**	Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) **deadtime** *minutes* **Example:** switch(config-radius)# **deadtime 30**	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) **server** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** switch(config-radius)# **server 10.10.1.1**	Configures the RADIUS server as a member of the RADIUS server group. **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* **Example:** switch(config-radius)# **use-vrf vrf1**	Specifies the VRF to use to contact the servers in the server group.
Step 7	**exit** **Example:** switch(config-radius)# **exit** switch(config)#	Exits configuration mode.

	Command or Action	Purpose
Step 8	(Optional) **show radius-server groups** [*group-name*] **Example:** `switch(config)# ` **`show radius-server groups`**	Displays the RADIUS server group configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# ` **`copy running-config startup-config`**	Copies the running configuration to the startup configuration.

Related Topics

Configuring the RADIUS Dead-Time Interval

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 NX-OS software uses any available interface.

Procedure

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

Related Topics

Configuring RADIUS Server Groups

Allowing Users to Specify a RADIUS Server at Login

By default, the NX-OS device forwards an authentication request based on the default AAA authentication method. You can configure the 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 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server directed-request**
	
	Example:
	
	`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	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

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

	Command or Action	**Purpose**
Step 6	(Optional) **show radius-server directed-request** **Example:** switch# **show radius-server directed-request**	Displays the directed request configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** 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 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 NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**radius-server retransmit** *count* **Example:** switch(config)# **radius-server retransmit 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* **Example:** 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	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** switch(config)# **show radius pending**	Displays the RADIUS configuration pending for distribution.
Step 5	(Optional) **radius commit** **Example:** switch(config)# **radius commit**	Applies the RADIUS configuration changes in the temporary database to the running configuration.

	Command or Action	Purpose

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

Step 7	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

Step 8	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`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 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 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **retransmit** *count*
	
	Example:
	
	`switch(config)# radius-server host`
	`server1 retransmit 3`
		Specifies the retransmission count for a specific server. The default is the global value.
		
		Note The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.

Step 3	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* **Example:**	Specifies the transmission timeout interval for a specific server. The default is the global value.

	Command or Action	Purpose
	switch(config)# **radius-server host server1 timeout 10**	**Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** switch(config)# **show radius pending**	Displays the RADIUS configuration pending for distribution.
Step 5	(Optional) **radius commit** **Example:** switch(config)# **radius commit**	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes RADIUS configuration to other NX-OS devices if you have enabled CFS configuration distribution for the user role feature.
Step 6	**exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 7	(Optional) **show radius-server** **Example:** switch# **show radius-server**	Displays the RADIUS server configuration.
Step 8	(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 Hosts

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** **Example:** `switch# ` **`configure terminal`** `switch(config)#`	Enters global configuration mode.
Step 2	(Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **acct-port** *udp-port* **Example:** `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** **Example:** `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* **Example:** `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** **Example:** `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	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# ` **`show radius pending`**	Displays the RADIUS configuration pending for distribution.
Step 7	(Optional) **radius commit** **Example:** `switch(config)# ` **`radius commit`**	Applies the RADIUS configuration changes in the temporary database to the running configuration.
Step 8	**exit** **Example:** `switch(config)# ` **`exit`** `switch#`	Exits configuration mode.
Step 9	(Optional) **show radius-server** **Example:**	Displays the RADIUS server configuration.

	Command or Action	Purpose
	switch(config)# **show radius-server**	
Step 10	(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 Hosts

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

	Command or Action	Purpose
Step 2	**radius-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `switch(config)# radius-server test username user1 password Ur2Gd2BH 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. **Note** For periodic RADIUS server monitoring, the idle timer value must be greater than 0.
Step 3	**radius-server deadtime** *minutes* **Example:** `switch(config)# radius-server deadtime 5`	Specifies the number of minutes before the 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.

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

Step 5	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring Periodic RADIUS Server Monitoring on Individual Servers

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 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 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**
	
	Example:
	
	`switch# configure terminal`
	`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*]]}
	
	Example:
	
	`switch(config)# radius-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.
		
		Note For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

Step 3	**radius-server deadtime** *minutes*
	
	Example:
	
	`switch(config)# radius-server deadtime`
	`5`
		Specifies the number of minutes before the 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**
	
	Example:
	
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

	Command or Action	Purpose

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

Related Topics

Configuring RADIUS Server Hosts

Configuring Global Periodic RADIUS Server Monitoring

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

Step 2	**radius-server deadtime** *minutes*
	
	Example:
	
	`switch(config)# radius-server deadtime`
	`5`
		Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.

Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.
Step 5	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 6	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

Step 7	(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]

Configuring One-Time Passwords

One-time password (OTP) support is available for NX-OS devices through the use of RSA SecurID token servers. With this feature, users authenticate to a 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

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

Before you begin

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

Ensure that the following are installed:

- 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 NX-OS device to support one-time passwords. However, you must configure the 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* **Example:** ```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* **Example:** ```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** **Example:** ```switch# configure terminal switch(config)#```	Enters global configuration mode.
Step 2	**aaa server radius dynamic-author** **Example:** ```switch(config)# aaa server radius dynamic-author```	Enables the RADIUS dynamic author server. You can disable the RADIUS dynamic author server using the no form of this command.

Configuring RADIUS Change of Authorization

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** ```switch# configure terminal switch(config)#```	Enters global configuration mode.
Step 2	[**no**] **aaa server radius dynamic-author** **Example:**	Configures the switch as an AAA server to facilitate interaction with an external policy server. You can disable the RADIUS dynamic

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

Verifying 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 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 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*} **Example:** `switch# ` **`show radius-server statistics`** **`10.10.1.1`**	Displays the RADIUS statistics.

Related Topics

Configuring RADIUS Server Hosts

Clearing RADIUS Server Statistics

Clearing RADIUS Server Statistics

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

Before you begin

Configure RADIUS servers on the NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	(Optional) **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** ```switch# show radius-server statistics 10.10.1.1```	Displays the RADIUS server statistics on the NX-OS device.
Step 2	**clear radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** ```switch# clear radius-server statistics 10.10.1.1```	Clears the RADIUS server statistics.

Related Topics

[Configuring RADIUS Server Hosts](#)

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
NX-OS Licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 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 TACACS+

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

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the 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 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 NX-OS device establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.

2. The NX-OS device will eventually receive one of the following responses from the TACACS+ daemon:
   ACCEPT
   User authentication succeeds and service begins. If the 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 NX-OS device. If the NX-OS device receives an ERROR response, the 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 NX-OS device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the 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 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 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 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 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 NX-OS device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A 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 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.
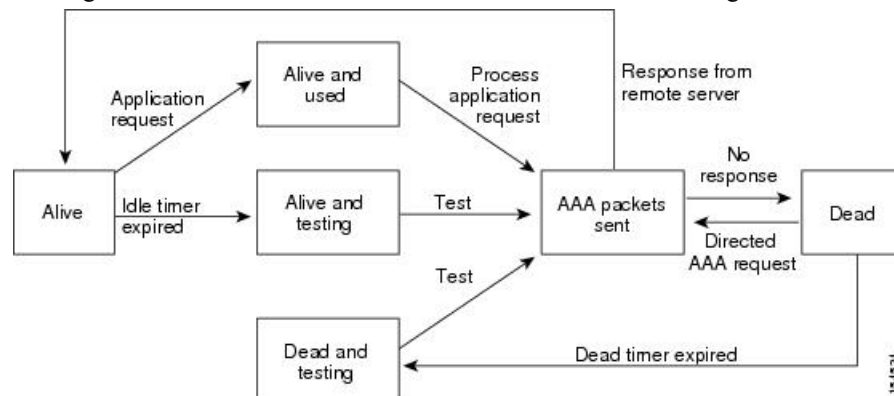
Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

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 TACACS+ implementation supports one vendor-specific option using the format recommended in the IETF specification. The 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 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 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 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 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 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 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).

Licensing Requirements for TACACS+

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	TACACS+ 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 NX-OS licensing scheme, see the NX-OS Licensing Guide.

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 NX-OS device.

- If you have a user account configured on the local NX-OS device that has the same name as a remote user account on an AAA server, the 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.

- 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 only for non-console sessions. If you use a console to login to the server, command authorization is disabled.=

Default Settings for TACACS+

This table lists the default settings for TACACS+ parameters.

Table 8: 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 NX-OS device.

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

TACACS+ Server Configuration Process

Procedure

Step 1 Enable TACACS+.

Step 2 Establish the TACACS+ server connections to the 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+

Enabling TACACS+

By default, the TACACS+ feature is disabled on the 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature tacacs+** **Example:** `switch(config)# feature tacacs+`	Enables TACACS+.

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

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

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

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

Related Topics

Enabling TACACS+

Configuring TACACS+ Server Groups

Configuring Global TACACS+ Keys

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

	Command or Action	**Purpose**

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

Enabling TACACS+

About AES Password Encryption and Master Encryption Keys

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

Step 2	**tacacs-server host** {*ipv4-address*|*ipv6-address* | *host-name*} **key** [**0** | **6** | **7**] *key-value*
	
	Example:
	`switch(config)# tacacs-server host`
	`10.10.1.1 key 0 PlIjUhYg`
		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 NX-OS software encrypts a clear text key before saving it to the running configuration. The default

	Command or Action	Purpose
	Example: 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+.
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 Master Encryption Keys

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

Step 2	**aaa group server tacacs+** *group-name*
	
	Example:
	
	`switch(config)# aaa group server tacacs+`
	`TacServer`
	`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*} **Example:** `switch(config-tacacs+)# server 10.10.2.2`	Configures the TACACS+ server as a member of the TACACS+ server group. If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.

Step 4	**exit**
	
	Example:
	
	`switch(config-tacacs+)# exit`
	`switch(config)#`
		Exits TACACS+ server group configuration mode.

Step 5	(Optional) **show tacacs-server groups** **Example:** `switch(config)# show tacacs-server groups`	Displays the TACACS+ server group configuration.

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

Related Topics

Enabling TACACS+

Remote AAA Services

Configuring TACACS+ Server Hosts

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 NX-OS software uses any available interface.

Procedure

	Command or Action	Purpose

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

Step 2	**ip tacacs source-interface** *interface*
	
	Example:
	
	`switch(config)# ip tacacs`
	`source-interface mgmt 0`
		Configures the global source interface for all TACACS+ server groups configured on the device.

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

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

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

Related Topics

> Enabling TACACS+
>
> Configuring TACACS+ Server Groups

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

Step 2	**tacacs-server directed-request**
	
	Example:
	
	`switch(config)# tacacs-server`
	`directed-request`
		Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.

Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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

Step 6	(Optional) **show tacacs-server directed-request**
	
	Example:
	
	`switch# show tacacs-server`
	`directed-request`
		Displays the TACACS+ directed request configuration.

Step 7	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+

Configuring the Timeout Interval for a TACACS+ Server

You can set a timeout interval that the NX-OS device waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the 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** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* **Example:** switch(config)# **tacacs-server host server1 timeout 10**	Specifies the timeout interval for a specific server. The default is the global value. **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**} **Example:** switch(config)# **show tacacs+ pending**	Displays the TACACS+ configuration pending for distribution.
Step 4	(Optional) **tacacs+ commit** **Example:** switch(config)# **tacacs+ commit**	Applies the TACACS+ configuration changes in the temporary database to the running configuration.
Step 5	**exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 6	(Optional) **show tacacs-server** **Example:** switch# **show tacacs-server**	Displays the TACACS+ server configuration.

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

Related Topics

[Enabling TACACS+](#)

Configuring TCP Ports

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

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose

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

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port*
	
	Example:
	
	`switch(config)# tacacs-server host`
	`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**}
	
	Example:
	
	`switch(config)# show tacacs+ distribution`
	` pending`
		Displays the TACACS+ configuration pending for distribution.

Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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

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

Related Topics

Enabling TACACS+

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 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** **Example:** switch# **configure terminal** 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*]]} **Example:** switch(config)# **tacacs-server test** **username user1 password Ur2Gd2BH** **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. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	**tacacs-server dead-time** *minutes* **Example:** switch(config)# **tacacs-server dead-time** **5**	Specifies the number of minutes before the 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** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 5	(Optional) **show tacacs-server** **Example:** switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch# **copy running-config** **startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Configuring Periodic TACACS+ Server Monitoring on Individual Servers

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 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**
	
	Example:
	
	`switch# configure terminal`
	`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*]]}
	
	Example:
	
	`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.
		
		Note For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.

Step 3	**tacacs-server dead-time** *minutes*
	
	Example:
	
	`switch(config)# tacacs-server dead-time`
	`5`
		Specifies the number of minutes before the 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** **Example:** `switch(config)# `**`exit`** `switch#`	Exits configuration mode.
Step 5	(Optional) **show tacacs-server** **Example:** `switch# `**`show tacacs-server`**	Displays the TACACS+ server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# `**`copy running-config`** **`startup-config`**	Copies the running configuration to the startup configuration.

Related Topics

Configuring TACACS+ Server Hosts

Configuring Global Periodic TACACS+ Server Monitoring

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 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** **Example:** `switch# `**`configure terminal`** `switch(config)#`	Enters global configuration mode.
Step 2	**tacacs-server deadtime** *minutes* **Example:** `switch(config)# `**`tacacs-server deadtime`** **`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**} **Example:** switch(config)# **show tacacs+ pending**	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** switch(config)# **tacacs+ commit**	Applies the TACACS+ configuration changes in the temporary database to the running configuration.
Step 5	**exit** **Example:** switch(config)# **exit** switch#	Exits configuration mode.
Step 6	(Optional) **show tacacs-server** **Example:** switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring ASCII Authentication

You can enable ASCII authentication on the TACACS+ server.

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**aaa authentication login ascii-authentication** **Example:** switch(config)# **aaa authentication login ascii-authentication**	Enables ASCII authentication. The default is disabled.

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

Configuring AAA Authorization on TACACS+ Servers

You can configure the default AAA authorization method for TACACS+ servers.

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**aaa authorization ssh-certificate default** {**group** *group-list* [**none**]	**local**	**none**} **Example:** switch(config)# **aaa authorization ssh-certificate** default group TACACSServer1 TACACSServer2	Configures the default AAA authorization method for the TACACS+ servers. The **ssh-certificate** keyword configures TACACS+ or local authorization with certificate authentication. The default authorization is local authorization, which is

	Command or Action	Purpose
		the list of authorized commands for the user's assigned role.
		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, and the **none** method specifies that no AAA authorization be used.

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

Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.

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

Related Topics

Enabling TACACS+

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 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** **Example:** ``` switch# configure terminal switch(config)# ```	Enters global configuration mode.
Step 2	**aaa authorization** {**commands** | **config-commands**} {**console** | **default**} {**group** *group-list* [**local**] | **local**} **Example:** ``` switch(config)# aaa authorization commands default group TacGroup Per command authorization will disable RBAC for all users. Proceed (y/n)? ```	Configures the command authorization method for specific roles on a TACACS+ server. The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword configures authorization sources for all configuration commands. The **console** keyword configures command authorization for a console session, and the **default** keyword configures command authorization for a non-console session. 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. 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**. If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond. If you press **Enter** at the confirmation prompt, the default action is **n**.

	Command or Action	Purpose
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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

Step 6	(Optional) **show aaa authorization** [**all**] **Example:** `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** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+

Testing Command Authorization on TACACS+ Servers

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*
	
	Example:
	`switch# test aaa authorization`
	`command-type commands`
	`user TestUser command reload`
		Tests a user's authorization for a command on the TACACS+ servers.
		
		The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands.
		
		Note Put double quotes (") before and after the *command-string* argument if it contains spaces.

Related Topics

> Enabling TACACS+
>
> Configuring Command Authorization on TACACS+ Servers
>
> Configuring User Accounts and RBAC

Enabling and Disabling Command Authorization Verification

You can enable and disable command authorization verificaiton 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*] **Example:** `switch# terminal verify-only`	Enables command authorization verification. After you enter this command, the NX-OS software indicates whether the commands you enter are authorized or not.
Step 2	**terminal no verify-only** [**username** *username*] **Example:** `switch# terminal no verify-only`	Disables command authorization verification.

Configuring Privilege Level Support for Authorization on TACACS+ Servers

You can configure privilege level support for authorization on TACACS+ servers.

Unlike IOS devices, which use privilege levels to determine authorization, Cisco NX-OS devices use role-based access control (RBAC). To enable both types of devices to be administered by the same TACACS+ servers, you can map the privilege levels configured on TACACS+ servers to user roles configured on NX-OS devices.

When a user authenticates with a TACACS+ server, the privilege level is obtained and used to form a local user role name of the format "priv-*n*," where *n* is the privilege level. The user assumes the permissions of this local role. Sixteen privilege levels, which map directly to corresponding user roles, are available. The following table shows the user role permissions that correspond to each privilege level.

Privilege Level	User Role Permissions
15	network-admin permissions
13 - 1	• Standalone role permissions, if the **feature privilege** command is disabled. • Same permissions as privilege level 0 with cumulative privileges for roles, if the **feature privilege** command is enabled.
0	Permission to execute **show** commands and **exec** commands (such as **ping**, **trace**, and **ssh**).

Important

Only the network administrator can escalate privileges to the root. As per the new security measures, a network operator (priv-1 user) is not allowed to collect show tech. Therefore, the enable command does not help to escalate the privileges.

Note

• When the **feature privilege** command is enabled, privilege roles inherit the permissions of lower level privilege roles.

• You must also configure the privilege level for the NX-OS device on the Secure Access Control Server (ACS).=

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature privilege** **Example:** `switch(config)# feature privilege`	Enables or disables the cumulative privilege of roles. Users can see the **enable** command only if this feature is enabled. The default is disabled.

Step 3	[**no**] **enable secret** [**0** | **5**] *password* [**priv-lvl** *priv-lvl* | **all**]
	
	Example:
	`switch(config)# enable secret 5 def456`
	`priv-lvl 15`
		Enables or disables a secret password for a specific privilege level. Users are prompted to enter the correct password upon each privilege level escalation. The default is disabled.

	Command or Action	Purpose
		You can enter **0** to specify that the password is in clear text or **5** to specify that the password is in encrypted format. The *password* argument can be up to 64 alphanumeric characters. The *priv-lvl* argument is from 1 to 15.
		Note To enable the secret password, you must have enabled the cumulative privilege of roles by entering the **feature privilege** command.
Step 4	[**no**] **username** *username* **priv-lvl** *n* **Example:** `switch(config)# username user2 priv-lvl 15`	Enables or disables a user to use privilege levels for authorization. The default is disabled. The **priv-lvl** keyword specifies the privilege level to which the user is assigned. There is no default privilege level. Privilege levels 0 to 15 (priv-lvl 0 to priv-lvl 15) map to user roles priv-0 to priv-15.
Step 5	(Optional) **show privilege** **Example:** `switch(config)# show privilege`	Displays the username, current privilege level, and status of cumulative privilege support.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Step 7	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 8	**enable** *level* **Example:** `switch# enable 15`	Enables a user to move to a higher privilege level. This command prompts for the secret password. The *level* argument specifies the privilege level to which the user is granted access. The only available level is 15.

Related Topics

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

Step 2	[**no**] **role name priv-***n*
	
	Example:
	
	`switch(config)# role name priv-5`
	`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* **Example:** `switch(config-role)# rule 2 permit 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. The *command-string* argument can contain spaces. **Note** Repeat this command for as many rules as needed.

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

Step 5	(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 Privilege Level Support for Authorization on TACACS+ Servers

Creating User Roles and Rules

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* **Example:** `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* **Example:** `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

Configuring TACACS+ Server Groups

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

Step 2	**no feature tacacs+** **Example:**	Disables TACACS+.

	Command or Action	Purpose
	`switch(config)# no feature tacacs+`	

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

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

Monitoring TACACS+ Servers

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

Before you begin

Configure TACACS+ servers on the NX-OS device.

Procedure

	Command or Action	Purpose

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

Related Topics

Configuring TACACS+ Server Hosts

Clearing TACACS+ Server Statistics

Clearing TACACS+ Server Statistics

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

Before you begin

Configure TACACS+ servers on the NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	(Optional) **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
	
	Example:
	`switch# show tacacs-server statistics`
	`10.10.1.1`
		Displays the TACACS+ server statistics on the NX-OS device.

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

Related Topics

[Configuring TACACS+ Server Hosts](#)

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+ 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.
show privilege	Displays the current privilege level, username, and status of cumulative privilege support.

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 enable the cumulative privilege of roles, configure a secret password for privilege level 2, and configure user3 for privilege level 2 authorization:

```
switch# configure terminal
switch(config)# feature privilege
switch(config)# enable secret def456 priv-lvl 2
switch(config)# username user3 priv-lvl 2
switch(config)# show privilege
User name: user3
Current privilege level: -2
Feature privilege: Enabled
switch(config)# copy running-config startup-config
switch(config)# exit
```

The following example shows how to change user3 from the priv-2 role to the priv-15 role. After entering the **enable 15** command, the user is prompted to enter the password that was configured by the administrator using the **enable secret** command. Privilege level 15 gives this user network-admin privileges under the enable mode.

```
User Access Verification
login: user3
Password: ******
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright ©) 2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch#
switch# enable 15
Password: def456
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright ©) 2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
```

```
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch-enable#
```

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
NX-OS licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 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.	—

C H A P T E R 6

Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on 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 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 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. 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 NX-OS device using LDAP, the following actions occur:

1. When the NX-OS device establishes a connection, it contacts the LDAP daemon to obtain the username and password.
2. The NX-OS device eventually receives one of the following responses from the LDAP daemon:

    • ACCEPT—User authentication succeeds and service begins. If the 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 NX-OS device. If the NX-OS device receives an ERROR response, the 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 NX-OS device. Users must first successfully complete LDAP authentication before proceeding to LDAP authorization.

3. If LDAP authorization is required, the 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 icon]

Note   In LDAP, authorization can occur before authentication.

LDAP Server Monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A 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 NX-OS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A 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 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 icon]
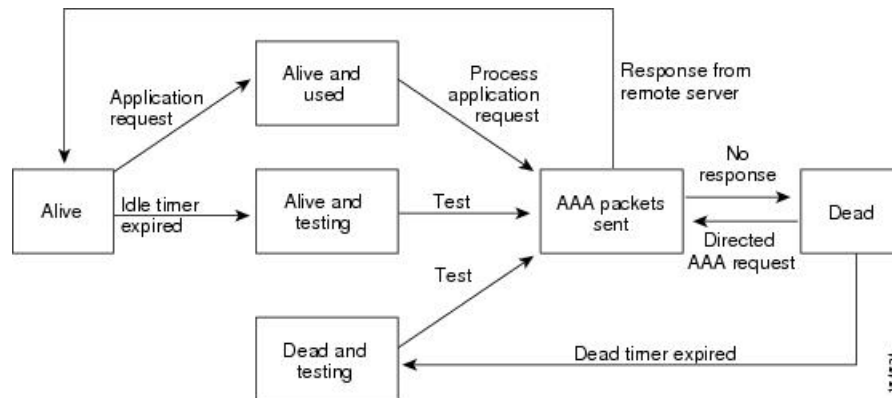
Note   The monitoring interval for alive servers and dead servers is different and can be configured by the user. The LDAP server monitoring is performed by sending a test authentication request to the LDAP server.

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.

VSA Format for LDAP

The LDAP implementation supports one vendor-specific option using the format recommended in the IETF specification. The 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 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 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 NX-OS software:

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

The 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 NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*.

Licensing Requirements for LDAP

The following table shows the licensing requirements for this feature:

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

Prerequisites for LDAP

LDAP has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
- Ensure that the 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.

- NX-OS supports only LDAP version 3.

- NX-OS supports only these LDAP servers:

    - OpenLDAP=

- Microsoft Active Directory

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

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

Default Settings for LDAP

This table lists the default settings for LDAP parameters.

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

Configuring LDAP

This section describes how to configure LDAP on a 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 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 NX-OS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

Procedure

	Command or Action	Purpose

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

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

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

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 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 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 NX-OS device.

Procedure

	Command or Action	Purpose

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

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

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

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

Related Topics

LDAP Server Configuration Process

Enabling or Disabling LDAP

Configuring LDAP Server Groups

Configuring the RootDN for an LDAP Server

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

Before you begin

Enable LDAP.

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

Procedure

	Command or Action	**Purpose**

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

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] | **timeout** *seconds*]]
	Example:
	`switch(config)# ldap-server host`
	`10.10.1.1 rootDN`
	`cn=manager,dc=acme,dc=com password`
	`Ur2Gd2BH timeout 60`
		Specifies the rootDN for the LDAP server database and the bind password for the root.
		
		Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.

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

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

Related Topics

LDAP Server Configuration Process

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 a member of this configured group in the LDAP server.

	Command or Action	Purpose
Step 6	(Optional) [**no**] **enable Cert-DN-match** **Example:** `switch(config-ldap)# enable Cert-DN-match`	Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login.
Step 7	(Optional) [**no**] **use-vrf** *vrf-name* **Example:** `switch(config-ldap)# use-vrf vrf1`	Specifies the VRF to use to contact the servers in the server group.

Step 8	**exit**
	
	Example:
	`switch(config-ldap)# exit`
	`switch(config)#`
		Exits LDAP server group configuration mode.

Step 9	(Optional) **show ldap-server groups** **Example:** `switch(config)# show ldap-server groups`	Displays the LDAP server group configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process

Configuring LDAP Server Hosts

Enabling or Disabling LDAP

Configuring LDAP Server Hosts

Configuring the Global LDAP Timeout Interval

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

Before you begin

Enable LDAP.

Procedure

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

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

Related Topics

Enabling or Disabling LDAP

Configuring the Timeout Interval for an LDAP Server

Configuring the Timeout Interval for an LDAP Server

Configuring the Timeout Interval for an LDAP Server

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

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds*
	Example:
	`switch(config)# ldap-server host server1`
	`  timeout 10`
		Specifies the timeout interval for a specific server. The default is the global value.
		
		Note The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.

	Command or Action	Purpose
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring TCP Ports

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

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

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

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

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

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

Related Topics

LDAP Server Configuration Process

Enabling or Disabling LDAP

Configuring LDAP Search Maps

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

Before you begin

Enable LDAP.

Procedure

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

	Command or Action	Purpose
	`switch(config-ldap-search-map)# exit` `switch(config)#`	
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

LDAP Server Configuration Process

Enabling or Disabling LDAP

Configuring Periodic LDAP Server Monitoring

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

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

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

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

Step 2	Required: [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]] **Example:**	Specifies the parameters for server monitoring. The default username is test, and the default password is Inspur. The default value for the idle timer is 60 minutes, and the valid range is from 1 to 1440 minutes.

	Command or Action	Purpose
	switch(config)# ldap-server host 10.10.1.1 test rootDN root1 username user1 password Ur2Gd2BH idle-time 3	**Note** We recommend that the user not be an existing user in the LDAP server database.
Step 3	[**no**] **ldap-server deadtime** *minutes* **Example:** switch(config)# ldap-server deadtime 5	Specifies the number of minutes before the 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** **Example:** switch(config)# show ldap-server	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process

Enabling or Disabling LDAP

Configuring LDAP Server Hosts

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 NX-OS device waits, after declaring that an LDAP server is dead, before sending out a test packet to determine if the server is now alive.

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

Before you begin

Enable LDAP.

Procedure

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

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

Related Topics

Enabling or Disabling LDAP

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

Step 2	**aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**} **Example:** `switch(config)# aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2`	Configures the default AAA authorization method for the LDAP servers. The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the **ssh-publickey** keyword configures LDAP or local authorization with the SSH public key. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are

	Command or Action	Purpose
		contacted for AAA authorization. The **local** method uses the local database for authorization.
Step 3	(Optional) **show aaa authorization** [**all**] **Example:** `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** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

> Enabling or Disabling LDAP

Monitoring LDAP Servers

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

Before you begin

Configure LDAP servers on the NX-OS device.

Procedure

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

Related Topics

> Configuring LDAP Server Hosts
> Clearing LDAP Server Statistics
> Clearing LDAP Server Statistics

Clearing LDAP Server Statistics

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

Before you begin

Configure LDAP servers on the NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	(Optional) **show ldap-server statistics** {*hostname*	*ipv4-address*	*ipv6-address*} **Example:** `switch# show ldap-server statistics 10.10.1.1`	Displays the LDAP server statistics.
Step 2	**clear ldap-server statistics** {*hostname*	*ipv4-address*	*ipv6-address*} **Example:** `switch# clear ldap-server statistics 10.10.1.1`	Clears the LDAP server statistics.

Related Topics

Monitoring LDAP Servers

Configuring LDAP Server Hosts

Monitoring LDAP Servers

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=inetOrgPerson)(cn=$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
```

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
NX-OS licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*

Standards

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

Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on 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 NX-OS device. SSH uses strong encryption for authentication. The SSH server in the 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 NX-OS device to make a secure, encrypted connection to another 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 NX-OS software works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the 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 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 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 device access. The SSH client is provided by 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 NX-OS device.

Licensing Requirements for SSH and Telnet

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	SSH and Telnet require 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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

- 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.

- 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.

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

Default Settings for SSH and Telnet

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

Table 9: 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** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**ssh key** {**dsa** [**force**]	**rsa** [*bits*[**force**]]	**ecdsa** [*bits* [**force**]]} **Example:** `switch(config)# ssh key rsa 2048`	Generates the SSH server key. The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048. The default value is 1024. You cannot specify the size of the DSA key. It is always set to 1024 bits. Use the **force** keyword to replace an existing key. **Note** If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all
Step 4	**ssh rekey max-data** *max-data* **max-time** *max-time*i **Example:** `switch(config)# ssh rekey max-data 1K` `max-time 1M`	Configures the rekey parameters.
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.
Step 6	**exit** **Example:** `switch(config)# exit` `switch#`	Exits global configuration mode.
Step 7	(Optional) **show ssh key** [**dsa**	**rsa**	**ecdsa**] [**md5**] **Example:** `switch# show ssh key`	Displays the SSH server keys. This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.

	Command or Action	Purpose
Step 8	**show run security all**	

Step 9	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`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_
	
	Example:
	
	`switch# copy`
	`tftp://10.10.1.1/secsh_file.pub`
	`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.

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

Step 3	**username** *username* **sshkey file bootflash:**_filename_
	
	Example:
	
	`switch(config)# username User1 sshkey`
	`file bootflash:secsh_file.pub`
		Configures the SSH public key in IETF SECSH format.

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

	Command or Action	Purpose
Step 5	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**username** *username* **sshkey** *ssh-key*
	
	Example:
	
	`switch(config)# username User1 sshkey`
	`ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZ19G+3flXswK3oiW4H7yUyuA50rv7gsEPj`
	`hOBfmsi6PAKui1nIf/DQhum+lNqJP/eLowb7uhO+1MRXFY/G+LNIQW3c9igG30c6k6+`
	`XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5S4Tplx8=`
		Configures the SSH public key in OpenSSH format.

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

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

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

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

Step 2	**ssh login-attempts** *number* **Example:** `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. **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** **Example:** `switch(config)# show running-config security all`	Displays the configured maximum number of SSH login attempts.
Step 4	(Optional) **copy running-config startup-config** **Example:** `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 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*] **Example:** `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*] **Example:** `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 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* **Example:** `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** **Example:** `switch(boot)# exit`	Exits boot mode.
Step 3	**copy scp:**//[*username@*]*hostname*/*filepath directory* **Example:**	Copies a file from the NX-OS device to a remote device using the Secure Copy Protocol (SCP). The default VRF is always used.

Command or Action	Purpose
```switch# copy scp://user1@10.10.1.1/users abc```	

Configuring SSH Passwordless File Copy

You can copy files from a 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**
	
	Example:
	```switch# configure terminal```
	```switch(config)#```
		Enters global configuration mode.

Step 2	[**no**] **username** *username* **keypair generate** {**rsa** [*bits* [**force**]] | **dsa** [**force**]}
	
	Example:
	```switch(config)# username user1 keypair```
	```generate rsa 2048 force```
		Generates the SSH public and private keys and stores them in the home directory ($HOME/.ssh) of the NX-OS device for the specified user. The NX-OS device uses the keys to communicate with the SSH server on the remote machine.
		
		The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048. The default value is 1024.
		
		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.

Step 3	(Optional) **show username** *username* **keypair**
	
	Example:
	```switch(config)# show username user1```
	```keypair```
		Displays the public key for the specified user.
		
		Note For security reasons, this command does not show the private key.

Step 4	Required: **username** *username* **keypair export** {**bootflash:***filename* | **volatile:***filename*} {**rsa** | **dsa**} [**force**]
	
	Example:
	```switch(config)# username user1 keypair```
	```export bootflash:key_rsa rsa```
		Exports the public and private keys from the home directory of the NX-OS device to the specified bootflash or volatile directory.
		
		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.
		
		To export the generated key pair, you are prompted to enter a passphrase that encrypts the private key. The private key is exported as

	Command or Action	Purpose
		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 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**]
	
	 Example:
	 `switch(config)# username user1 keypair`
	 `import bootflash:key_rsa rsa`
		Imports the exported public and private keys from the specified bootflash or volatile directory to the home directory of the NX-OS device.
		
		 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.
		
		 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 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 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 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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **feature scp-server** **Example:** switch(config)# feature scp-server	Enables or disables the SCP server on the NX-OS device.
Step 3	Required: [**no**] **feature sftp-server** **Example:** switch(config)# feature sftp-server	Enables or disables the SFTP server on the NX-OS device.
Step 4	Required: **exit** **Example:** switch(config)# exit switch#	Exits global configuration mode.
Step 5	(Optional) **show running-config security** **Example:** switch# show running-config security	Displays the configuration status of the SCP and SFTP servers.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

Step 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**}
	
	Example:
	
	`switch(config)# username jsmith`
	`ssh-cert-dn "/O = ABCcompany, OU = ABC1,`
	`emailAddress = jsmith@ABCcompany.com,`
	`L = Metropolis, ST = New York, C = US,`
	` CN = jsmith" rsa`
		Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 4	[**no**] **crypto ca trustpoint** *trustpoint*
	
	Example:
	
	`switch(config)# crypto ca trustpoint`
	`winca`
	`switch(config-trustpoint)#`
		Configures a trustpoint.
		
		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*	Configures a CA certificate for the trustpoint.

	Command or Action	Purpose
	Example: switch(config-trustpoint)# crypto ca authenticate winca	**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** **Example:** switch(config-trustpoint)# crypto ca 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). **Note** Static CRL is the only supported revocation check method. **Note** To delete the CRL, enter the **delete crl** command.
Step 7	(Optional) **show crypto ca certificates** **Example:** switch(config-trustpoint)# show crypto ca certificates	Displays the configured certificate chain and associated trustpoint.
Step 8	(Optional) **show crypto ca crl** *trustpoint* **Example:** switch(config-trustpoint)# show crypto ca crl winca	Displays the contents of the CRL list of the specified trustpoint.
Step 9	(Optional) **show user-account** **Example:** switch(config-trustpoint)# show user-account	Displays configured user account details.
Step 10	(Optional) **show users** **Example:** switch(config-trustpoint)# show users	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:** switch(config-trustpoint)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring Legacy SSH Algorithm Support

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

Procedure

	Command or Action	**Purpose**

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

Step 2	(Optional) **ssh kexalgos all** **Example:** `switch(config)# ssh kexalgos all`	Enables all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys. Supported KexAlgorithmns are: • curve25519-sha256 • diffie-hellman-group-exchange-sha256 • diffie-hellman-group14-sha1 • diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521
Step 3	(Optional) **ssh macs all** **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
Step 4	(Optional) **ssh ciphers all** **Example:** `switch(config)# ssh ciphers all`	Enables all supported ciphers to encrypt the connection. Supported ciphers are: • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes256-gcm@openssh.com

	Command or Action	Purpose
		• aes128-gcm@openssh.com
Step 5	(Optional) **ssh keytypes all** **Example:** `switch(config)# ssh keytypes all`	Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client. Supported key types are: • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521 • ssh-dss • ssh-rsa

Changing the Default SSH Server Port

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

Procedure

	Command or Action	Purpose

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

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

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

Step 4	**ssh port** *local-port* **Example:** `switch(config)# ssh port 58003`	Configures the port.

	Command or Action	Purpose
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

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

Step 7	(Optional) **show running-config security all** **Example:** `switch# ssh port 58003`	Displays the security configuration.

Step 8	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`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** **Example:** `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 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

Step 4	(Optional) **show ssh server** **Example:** `switch# show ssh server`	Displays the SSH server configuration.

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

Deleting SSH Server Keys

You can delete SSH server keys on the 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**no ssh key** [**dsa** | **rsa** | **ecdsa**] **Example:** `switch(config)# no ssh key rsa`	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	switch(config)# exit switch#	
Step 5	(Optional) **show ssh key** **Example:** switch# show ssh key	Displays the SSH server key configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Related Topics

Generating SSH Server Keys

Clearing SSH Sessions

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

Procedure

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

Configuring Telnet

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

Enabling the Telnet Server

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

Procedure

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

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	**feature telnet** **Example:** switch(config)# feature telnet	Enables the Telnet server. The default is disabled.
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits global configuration mode.
Step 4	(Optional) **show telnet server** **Example:** switch# show telnet server	Displays the Telnet server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config 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 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 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*] **Example:** 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*] **Example:** switch# telnet6 2001:0DB8::ABCD:1 vrf 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

Clearing Telnet Sessions

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

Before you begin

Enable the Telnet server on the NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `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. For 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.

Command	Purpose
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
**************************************
**************************************
```

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 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 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 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 NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the 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 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:

```
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
NX-OS licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*

RFCs

RFCs	Title
RFC 6187	*X.509v3 Certificates for Secure Shell Authentication*

Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the 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 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 NX-OS device can also enroll with a trust point to obtain an identity certificate to associate with a key pair. This trust point is called an *identity CA*.

RSA Key Pairs and Identity Certificates

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

The 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 NX-OS device trusts for peer certificate verification for any application (such as SSH).

- A 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 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 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 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 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 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 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 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 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 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 NX-OS device can verify peer certificates. The 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 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 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 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, 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 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 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 NX-OS software does not perform CRL checking and considers the certificate to be not revoked unless you have configured other revocation checking methods.

Import and Export Support for Certificates and Associated Key Pairs

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

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

Licensing Requirements for PKI

The following table shows the licensing requirements for this feature:

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

Guidelines and Limitations for PKI

PKI has the following configuration guidelines and limitations:

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

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

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

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

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

- Configuration rollbacks do not support the PKI configuration.

- The NX-OS software does not support OSCP.=

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

Default Settings for PKI

This table lists the default settings for PKI parameters.

Table 10: 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 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 NX-OS software uses the fully qualified domain name (FQDN) of the device as the subject in the identity certificate. Also, the NX-OS software uses the device FQDN as a default key label when you do not specify a label during key-pair generation. For example, a certificate named DeviceA.example.com is based on a device hostname of DeviceA and a device IP domain name of example.com.

⚠️

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

Procedure

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

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

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

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

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

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

Generating an RSA Key Pair

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

Beginning NX-OS Release 9.3(3), you must explicitly generate RSA key pairs before you associate the NX-OS device with a trust point CA. Prior to 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]
	
	Example:
	`switch(config)# crypto key generate rsa`
	` exportable`
		Generates an RSA key pair. The maximum number of key pairs on a device is 16.
		
		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, and 2048. The default modulus size is 512.
		Note The security policy on the NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus.
		By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format.
		Caution You cannot change the exportability of a key pair.

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

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

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

Creating a Trust Point CA Association

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

Before you begin

Generate the RSA key pair.

Procedure

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

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

Related Topics

Generating an RSA Key Pair

Authenticating the CA

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

✎

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

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	**Purpose**

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

Step 2	**crypto ca authenticate** *name*
	
	Example:
	`switch(config)# crypto ca authenticate`
	`admin-ca`
	`input (cut & paste) CA certificate`
	`(chain) in PEM format;`
	`end the input with a line containing only`
	`END OF INPUT :`
	`-----BEGIN CERTIFICATE-----`
	`MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRI1jK0ZejANBgkqhkiG9w0BAQUFADCB`
	`kDEgMB4GCSqGSIb3DQEJARYRWllhorRrzZLBjaXNjby5jb20xCzAJBgNVBAYTAklO`
	`MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhomdhbG9yZTIEOMAwGA1UE`
	`ChMFQ2lzY28xEzARBgNVBAsTCnf5ldHNb3JhZZUxEjAQBgNVBAMTCUFwYXUiYSBD`
	`QIPAeRw0wNTAlMDMyMjQ2MzdaFw0wNzAlMDMyMjU1MTdaMIIGQMSAwHgYJKoZIhvcN`
	`AQkEFhFhbWFtZGtlQGNpc2NvLmNvbIEIMAkGA1UEBHMCSU4xEjAQBgNVBAgTCUth`
	`crr6hnGGFrYIESMBAGA1UEB+MQqnFuZ2Fsb3JlIMQ4wDAYDVQQKEwVDaXNjozEIMBEG`
	`A1UECxMKbmV0c3RvcmFnZIESMBAGA1UEAxMQXBhomfShIENBMFwwDQYJKoZIhvcN`
	`AQEBBQADSwAwSAJBAMW/7o3+DXJFANBsIHHzluNcdM87ypyzwuoSNZXQMpeRXXI`
	`OzyBAgiXIZ2ASFUOwQ1iDM8rO/41jf8RxvVKvysCAwEAAaOBvzCBvDAIBgNVHQ8E`
	`BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUyjyRoMbrCNMRU2OyRrQ`
	`GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NtZS5mwOC9DZXJ0RW5yb2xs`
	`L0FwYXXUiYSUyMENBbmNyoDAwoO6gLIYc2mlsZIovLlxcc3NlLIPA4XEN1crREcnJv`
	`bGxcQXBhcrrShJTIwQ0BtiY3JSMBAGCSsGAQQBgrjcVAQQDAgEAMA0GCSqGSIb3DQEB`
	`BQUAA0EAHv6UQ+8rtE399Tiww+KeGr0gONLIaqNgIh0AFcITOrEyuyt/WGPzksF9Ea`
	`NBG7E0oN66zex0EOEfG1Vs6mXp1//w==`
	`-----END CERTIFICATE-----`
	`END OF INPUT`
	`Fingerprint(s): MD5`
	`Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12`
	`Do you accept this certificate? [yes/no]:`
	`yes`
		Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.
		
		The maximum number of trust points that you can authenticate to a specific CA is 10.
		
		Note For subordinate CA authentication, the NX-OS software requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export.

	Command or Action	Purpose

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

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

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

Related Topics

Creating a Trust Point CA Association

Configuring Certificate Revocation Checking Methods

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

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

Before you begin

Authenticate the CA.

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

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose
Step 3	**revocation-check** {**crl** [**none**] | **none**} **Example:** `switch(config-trustpoint)# revocation-check none`	Configures the certificate revocation checking methods. The default method is **crl**. The NX-OS software uses the certificate revocation methods in the order that you specify.

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

Step 5	(Optional) **show crypto ca trustpoints** **Example:** `switch(config)# show crypto ca trustpoints`	Displays the trust point CA information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Authenticating the CA

Configuring a CRL

Generating Certificate Requests

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

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	Purpose

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

	Command or Action	**Purpose**
Step 2	**crypto ca enroll** *name* **Example:** ``` switch(config)# crypto ca enroll admin-ca Create the certificate request .. Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a note of it. Password:nbv123 The subject name in the certificate will be: DeviceA.cisco.com Include the switch serial number in the subject name? [yes/no]: no Include an IP address in the subject name [yes/no]: yes ip address:172.22.31.162 The certificate request will be displayed... -----BEGIN CERTIFICATE REQUEST----- MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUU1DVaSMqNIgJ2kt8r141KY 0JC6ManNy4cpxk8VeMKZSiLJ4JgTIzKWdxbiDkTTysnjuCXGvjb+wj0Hhv/y51T9y P2NU8ornqShrvFZgC7ysN/PyMwKcgzhOVpj+rxarqZvHtGJ91XIq4WoVkSCzXv8S VCyHDVEvAgMAAGgJIzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDkGCSqGSIb3DQEJ DjEpMCcwJQYIVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ KoZIhvcNAQEBBQADgYEAkT60KER6Qo8rrj0sDKZMHSfJZh6K6JtDz3Gxd99GlFWgt PftrNdWUE/pw6HayfQl2T3ecqNwel2d15133YBF2bktExiI6JL88rITOjglXMjja8 8a23dNDpNsM8rklwA6rWkrVl8NUZEFJxqbrjfrngRNIZacJUUS6ZqKCMetdKytUk0= -----END CERTIFICATE REQUEST----- ```	Generates a certificate request for an authenticated CA. **Note** You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked.
Step 3	**exit** **Example:** ``` switch(config-trustpoint)# exit switch(config)# ```	Exits trust point configuration mode.
Step 4	(Optional) **show crypto ca certificates** **Example:** ``` switch(config)# show crypto ca certificates ```	Displays the CA certificates.
Step 5	(Optional) **copy running-config startup-config** **Example:** ``` switch(config)# copy running-config startup-config ```	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association

Installing Identity Certificates

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

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** ```switch# configure terminal\nswitch(config)#```	Enters global configuration mode.
Step 2	**crypto ca import** *name* **certificate** **Example:** ```switch(config)# crypto ca import admin-ca\n certificate\ninput (cut & paste) certificate in PEM\nformat:\n-----BEGIN CERTIFICATE-----\nMIIEADCCA6qgAwIBAgIKCjOOoQAAAAAoDANBgkqhkiG9w0BAQUFADCBkDEgMB4G\nCSqGSIb3DQEJARYRYWlhbmRrZLBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD\nVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUdhomdhoG9yZTEOMAwGA1UEChMEQ2lz\nY28xEzARBgNVBAsTCtr5ldHdlb3JrZJUEjAQBgNVBAMICUFwXJUySBDQIAeFw0w\nNIExMTIwMzAyNDBaFw0wNjExMTIwMZEyNDBaMBwxGjAYBgNVBAMIEVZlZZFzLIEu\nY2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNACdjQu4lC\ndQlWkjKjSICdpLfK5eJ3tNCQrjGpzuJKsZPFXjF2UoiyeCYE8ylndwWyw5E08rJ47\ngJxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWVAu8rDfz8jMChIM4WlaY/cp2q4Go\nx7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw\nGYIRVivhYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR00EBYEFKCLi+2sspWEfgrR\nbhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCccc8kaDG6wjIEMNjskYUBoLFmx+cYGW\npIGIMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtGtlQGNpc2NvLmNvbTELMAkGA1UEB\nBhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTYYIESMBAGA1UEBxMQJmFtZZFsb3J1MQ4w\nDAYDVQQKEwvDaXNjozEIMBEGA1UECxMKbmV0V0c3RvcmtFHZIESMBAGA1UEAxMJQKBh\ncm5hIENBghAFYNKJrlQ21E9UEiWMrR1dMGsGA1UdHwRkMGIwLqAsoCcGMGh0dHA6\nLy9zc2UtdMdgvQ2VydEVucm9sbOC9BcGFybnREIMjED0QS5jcnwwMKAuoCyGKmZpbGU6\nLy9cXHNzZS0wOEXIZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH\nAQEEfjB8MDoGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbJvbGwvc3NL\nLTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4\nXENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF\nAANBADbGBGsbe7GNLh9xeODWBNbm24U69ZSuDDcOdUZUUIgrqprIdjVpPyejtsyflw\nE36cIZu4WsExREqxbTk8ycx7V5o=\n-----END CERTIFICATE-----```	Prompts you to cut and paste the identity certificate for the CA named admin-ca. The maximum number of identify certificates that you can configure on a device is 16.
Step 3	**exit** **Example:** ```switch(config)# exit\nswitch#```	Exits configuration mode.

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

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

Related Topics

Creating a Trust Point CA Association

Ensuring Trust Point Configurations Persist Across Reboots

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

The trust point configuration is a normal 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

Exporting Identity Information in PKCS 12 Format

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

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

Before you begin

Authenticate the CA.

Install an identity certificate.

Procedure

	Command or Action	Purpose

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

Step 2	**crypto ca export** *name* **pkcs12 bootflash:***filename password*
	
	Example:
	`switch(config)# crypto ca export admin-ca`
	`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**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Related Topics

Generating an RSA Key Pair

Authenticating the CA

Installing Identity Certificates

Importing Identity Information in PKCS 12 Format

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

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

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

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

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

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

	Command or Action	Purpose

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

Configuring a CRL

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

Before you begin

Ensure that you have enabled certificate revocation checking.

Procedure

	Command or Action	Purpose

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

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

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

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

Step 5	(Optional) **show crypto ca crl** *name* **Example:**	Displays the CA CRL information.

	Command or Action	Purpose
	`switch# show crypto ca crl admin-ca`	
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting Certificates from the CA Configuration

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

Procedure

	Command or Action	Purpose

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

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

Step 3	**delete ca-certificate** **Example:** `switch(config-trustpoint)# delete ca-certificate`	Deletes the CA certificate or certificate chain.
Step 4	**delete certificate** [**force**] **Example:** `switch(config-trustpoint)# delete certificate`	Deletes the identity certificate. You must use the **force** option if the identity certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only the identity certificate and leave the applications (such as SSH) without a certificate to use.
Step 5	**exit** **Example:**	Exits trust point configuration mode.

	Command or Action	Purpose
	`switch(config-trustpoint)# exit` `switch(config)#`	

Step 6	(Optional) **show crypto ca certificates** [*name*]
	
	Example:
	`switch(config)# show crypto ca`
	`certificates admin-ca`
		Displays the CA certificate information.

Step 7	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Deleting RSA Key Pairs from a NX-OS Device

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

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

Procedure

	Command or Action	Purpose

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

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

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

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

	Command or Action	Purpose

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

[Generating Certificate Requests](#)

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 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 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 NX-OS Device

To configure certificates on a 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
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
```

```
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
Do you accept this certificate? [yes/no]:y

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

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

```
Device-1(config)# crypto ca enroll myCA
 Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password: nbv123
 The subject name in the certificate will be: Device-1.cisco.com
 Include the switch serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address: 10.10.1.1
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEBBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

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

Step 10   Import the identity certificate.

```
Device-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
```

```
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzACHi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Device-1(config)# exit
Device-1#
```

Step 11   Verify the certificate configuration.

Step 12   Save the certificate configuration to the startup configuration.

Related Topics

      Downloading a CA Certificate

      Requesting an Identity Certificate

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**.
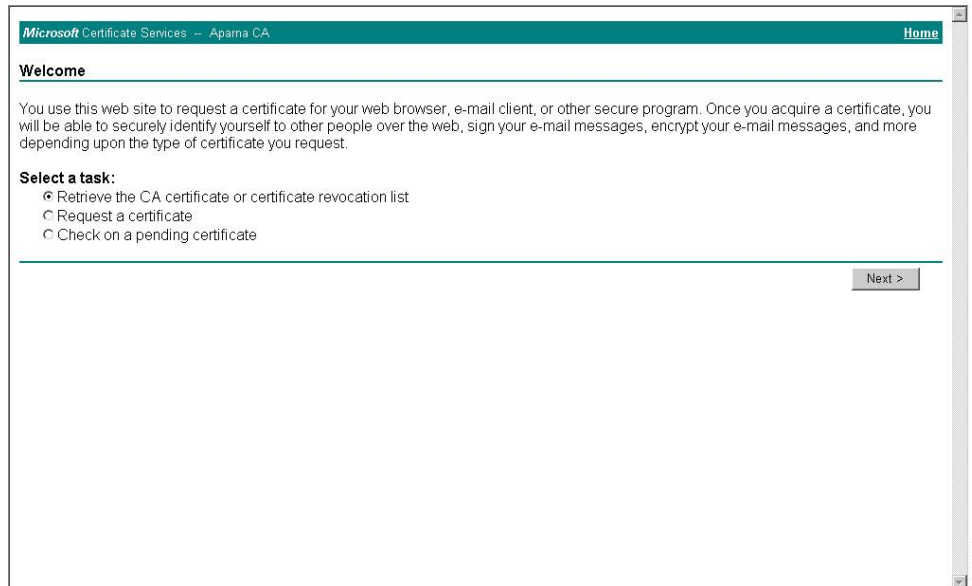
Step 2   From the display list, choose the CA certificate file to download from the displayed list. Then click **Base 64 encoded** and click **Download CA certificate**.
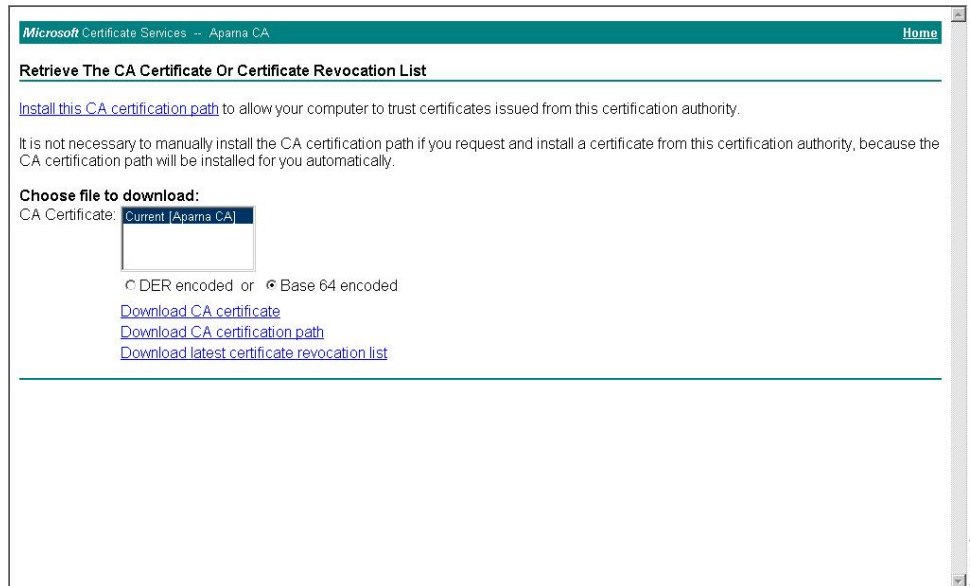
Step 3   Click **Open** in the File Download dialog box.
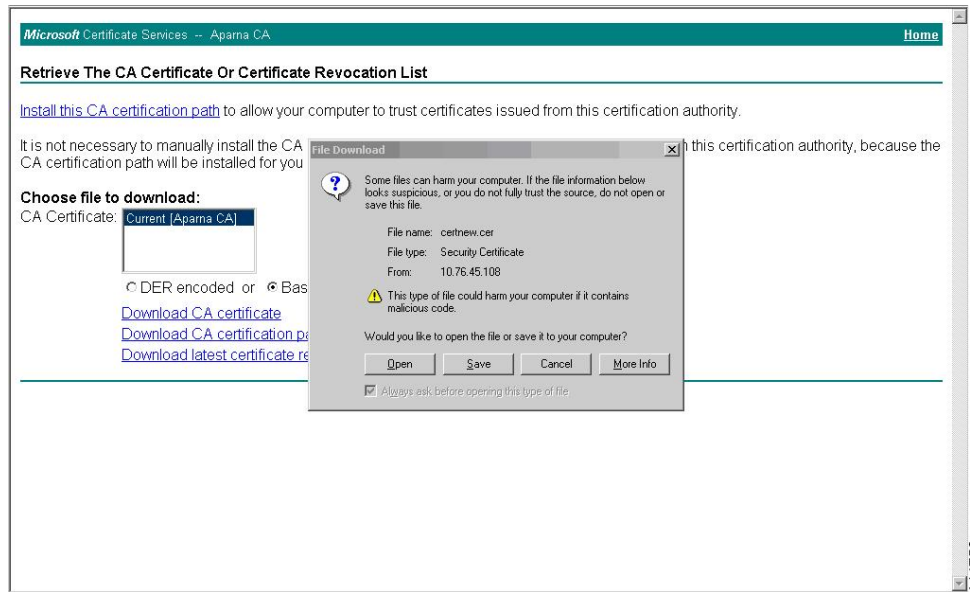
Step 4   In the Certificate dialog box, click **Copy to File** and click **OK**.
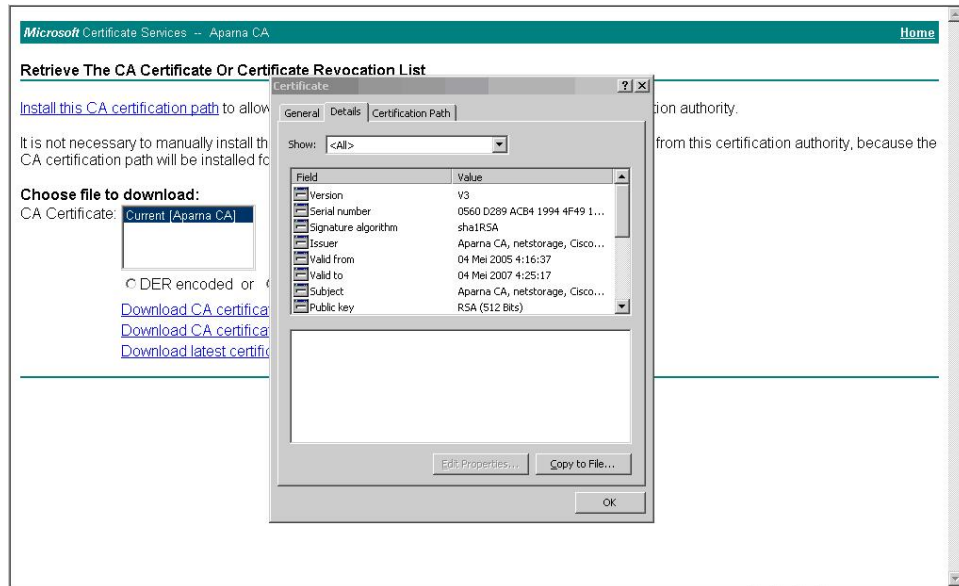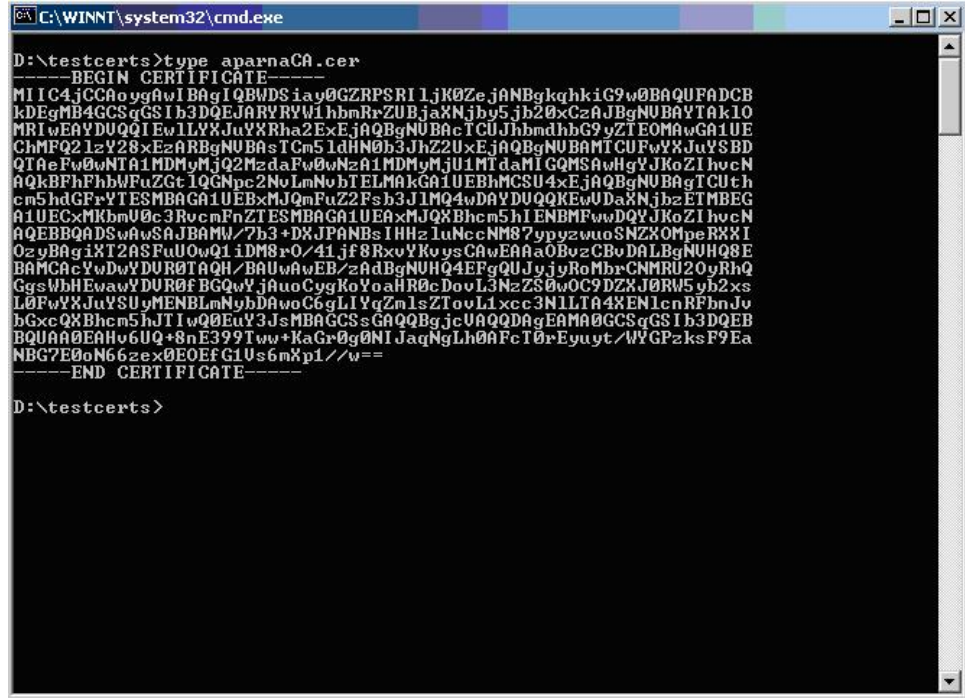
Step 5   From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next**.
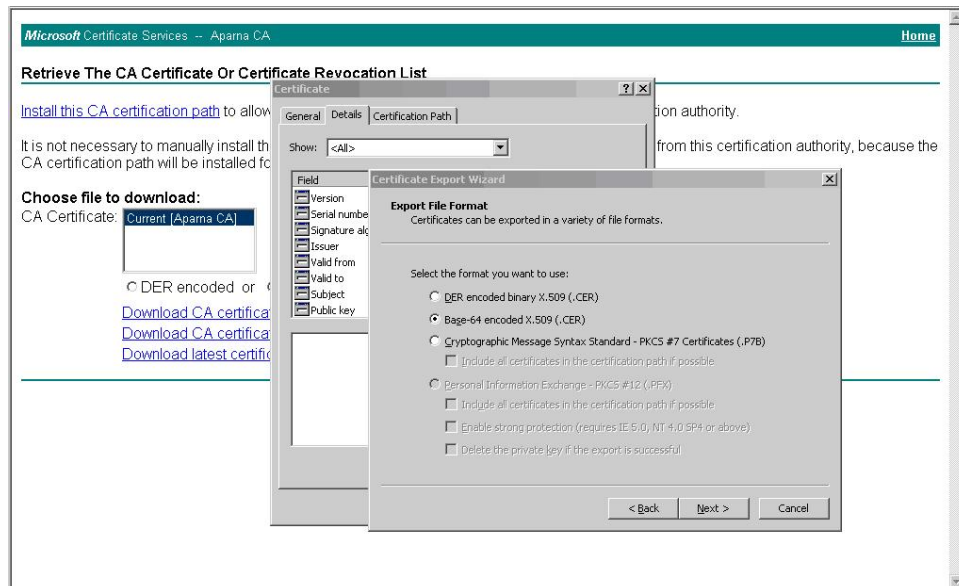
Step 6   In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.

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**.
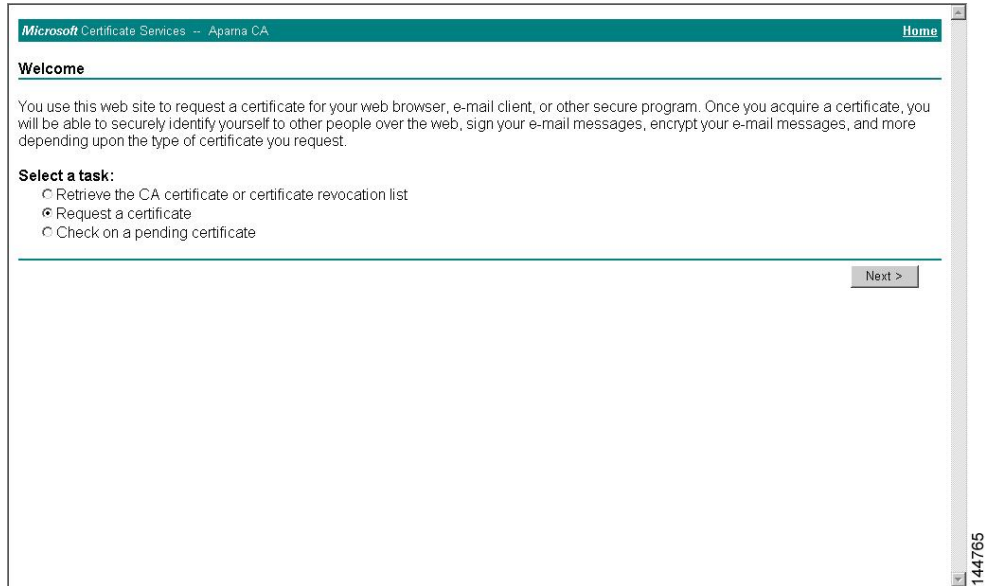
Step 2   Click **Advanced request** and click **Next**.
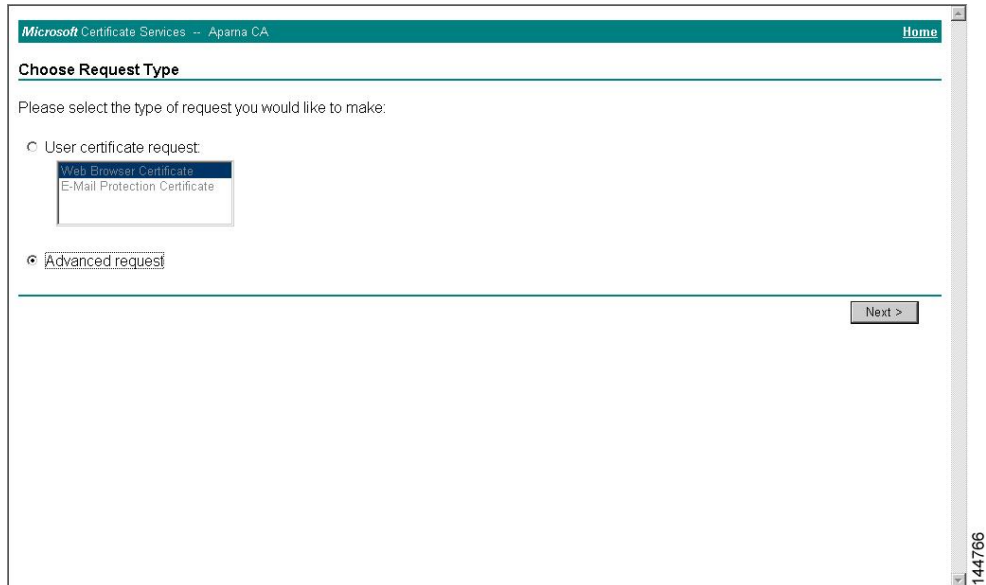
Step 3   Click **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** and click **Next**.
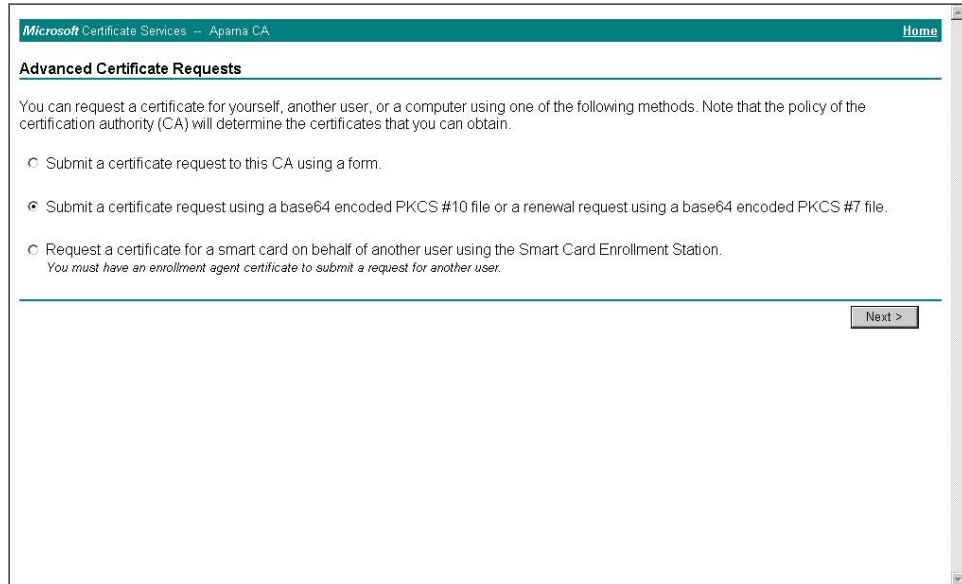
Step 4   In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next**. The certificate request is copied from the NX-OS device console.
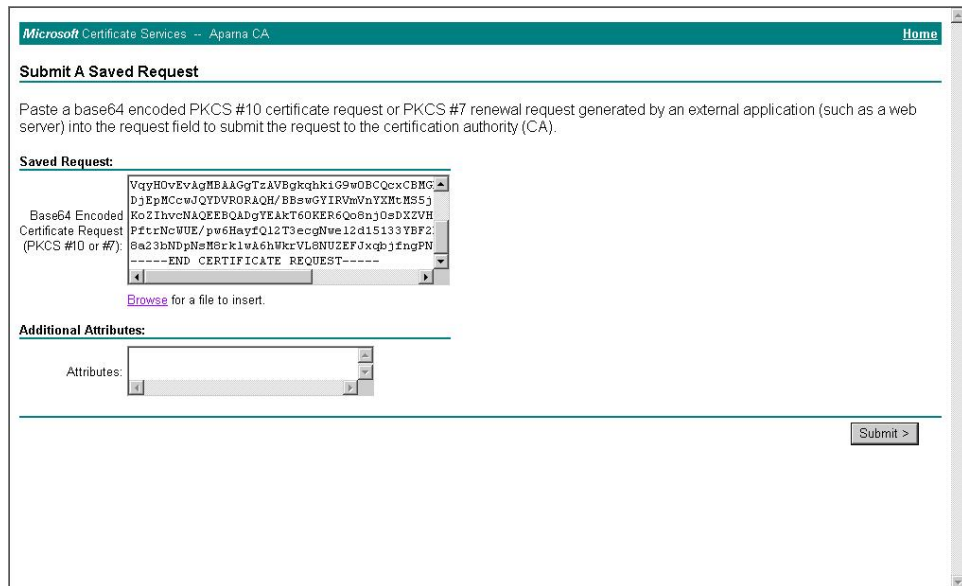
Step 5 Wait one or two days until the certificate is issued by the CA administrator.
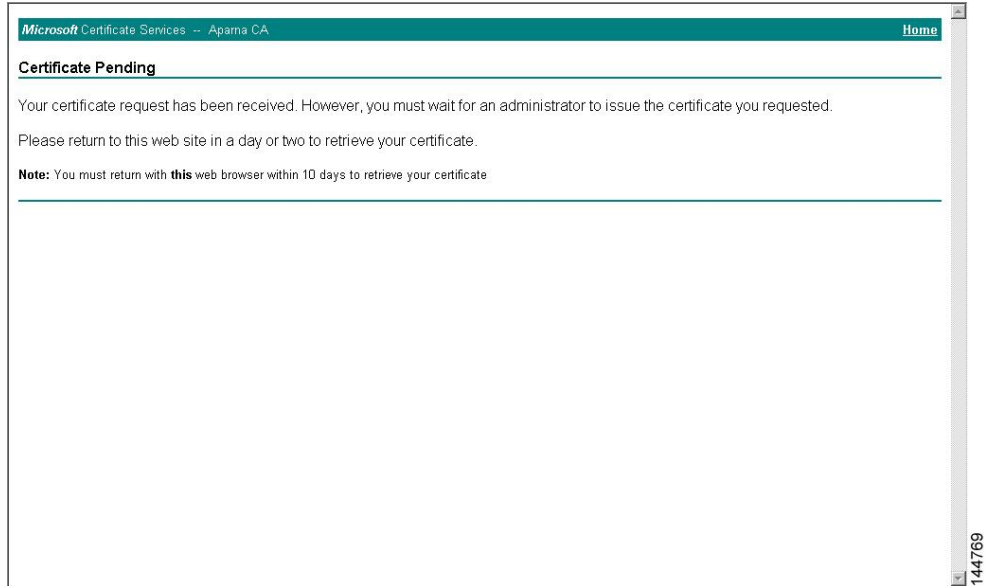
Step 6 Note that the CA administrator approves the certificate request.
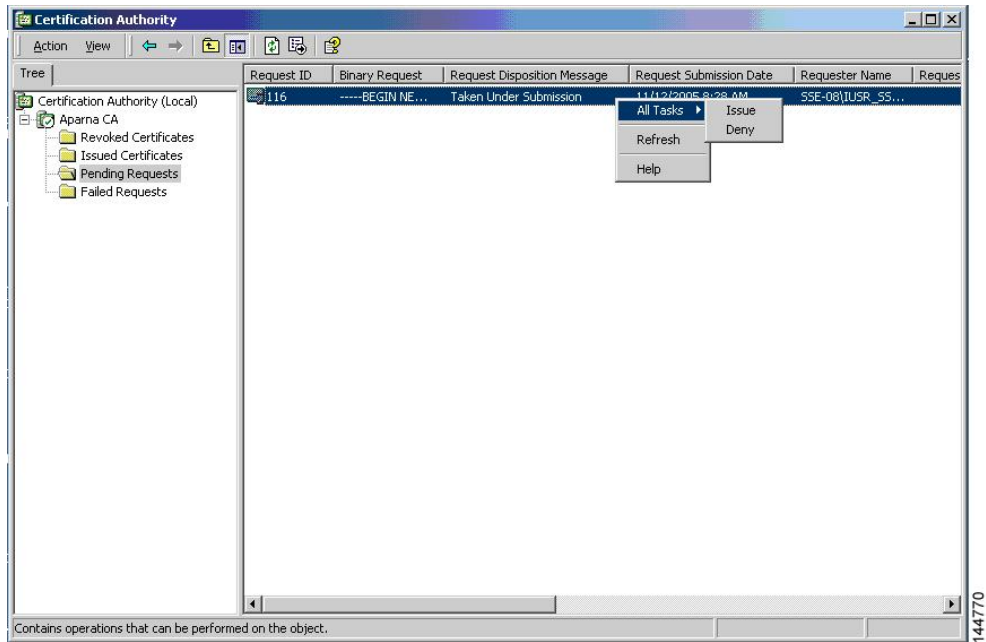
Step 7   From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**.
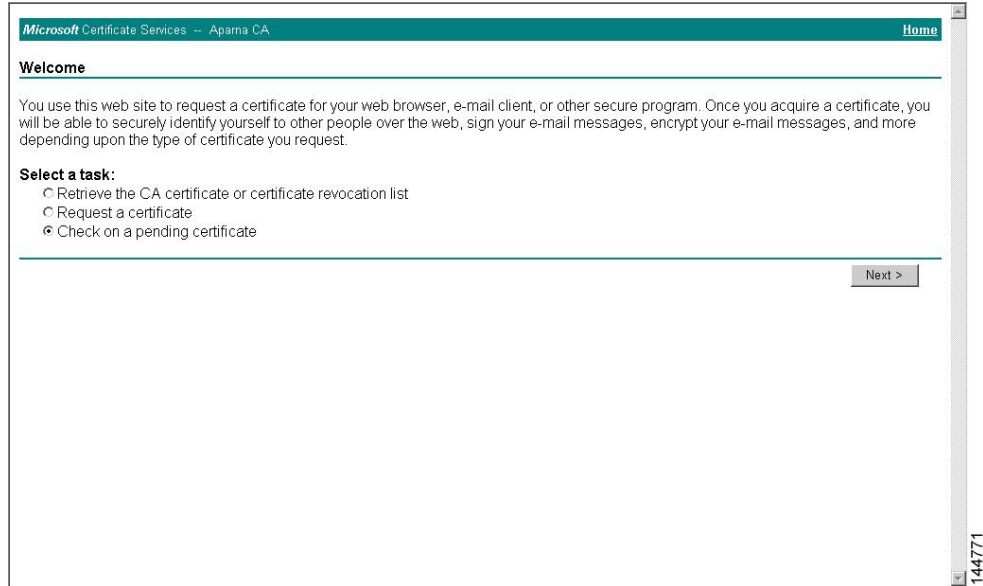
Step 8   Choose the certificate request that you want to check and click **Next**.
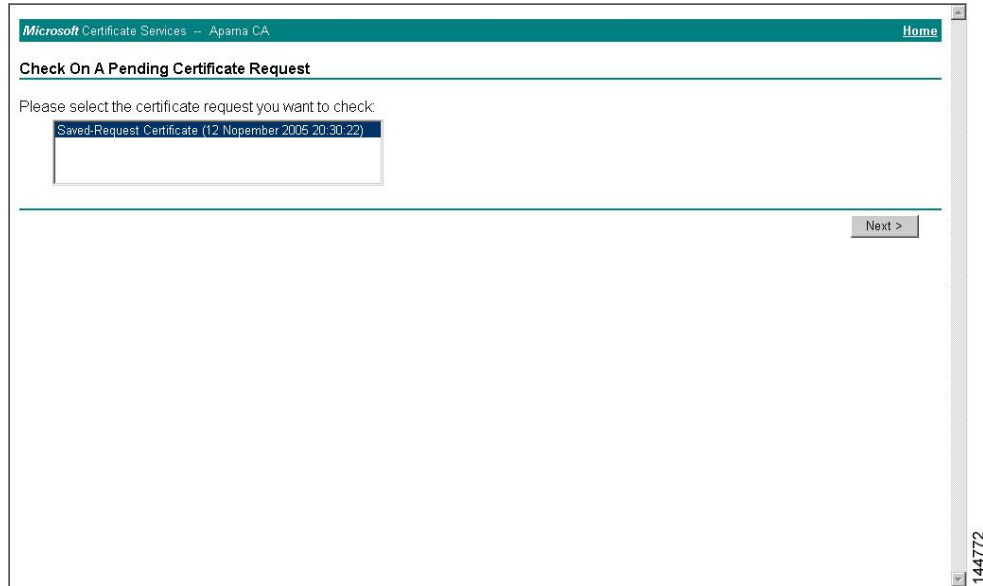
Step 9   Click **Base 64 encoded** and click **Download CA certificate**.
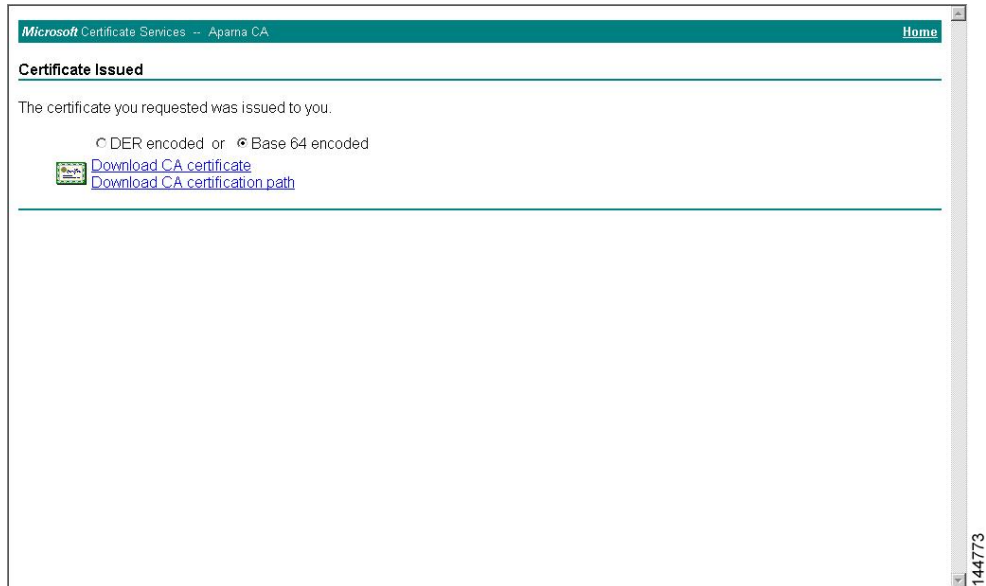
Step 10   In the File Download dialog box, click **Open**.
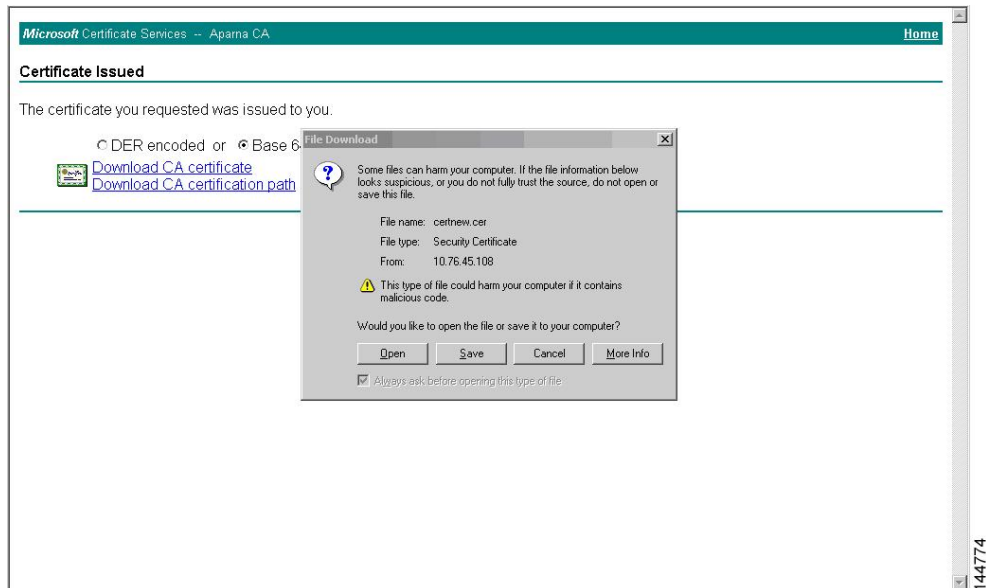
Step 11 In the Certificate box, click **Details** tab and click **Copy to File...**. In the Certificate Export Dialog box, click **Base-64 encoded X.509 (.CER)**, and click **Next**.
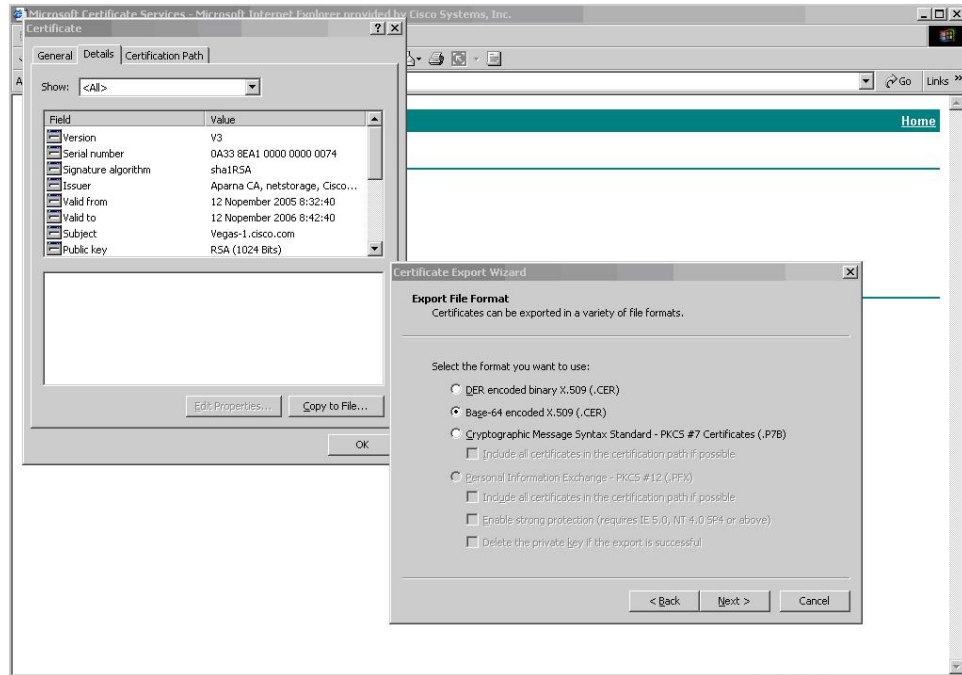
Step 12 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and
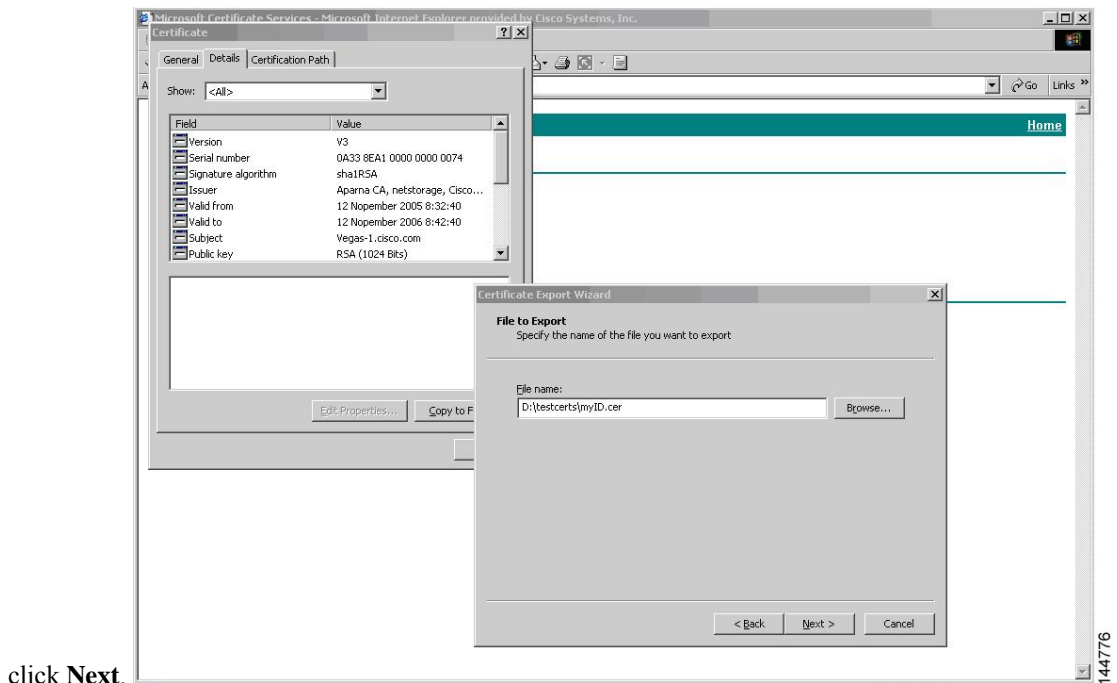
click **Next**.
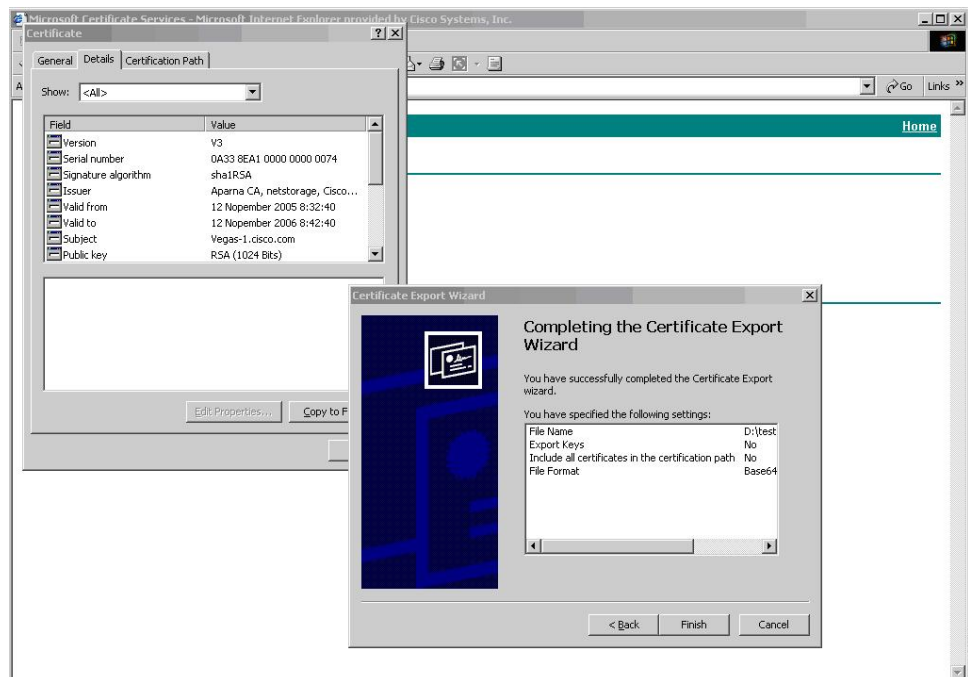
Step 13 Click **Finish**.

Step 14 Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.

Related Topics

Generating Certificate Requests

Configuring Certificates on a Cisco NX-OS Device

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**.
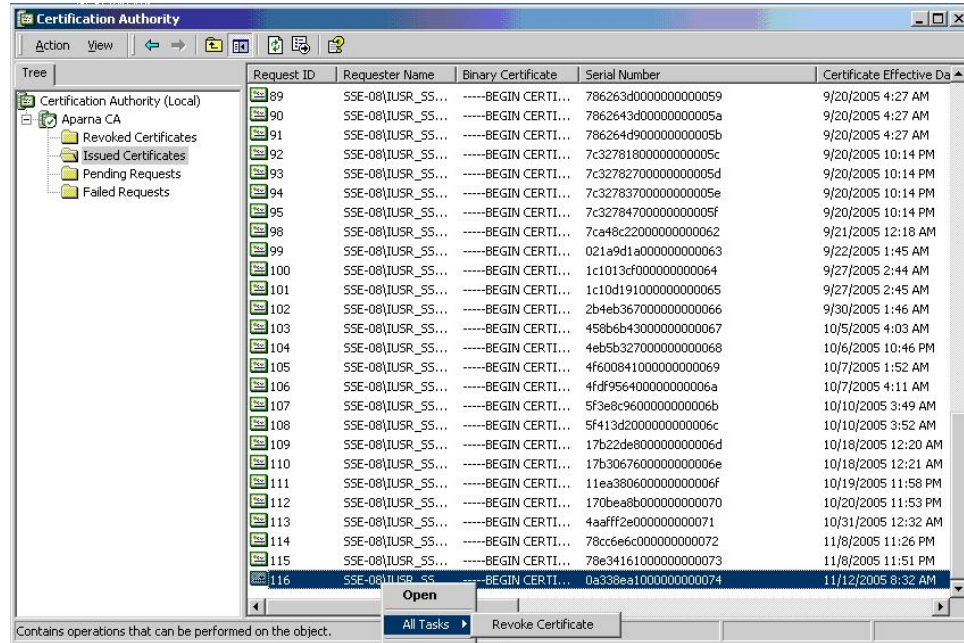
Step 3   From the Reason code drop-down list, choose a reason for the revocation and click **Yes**.
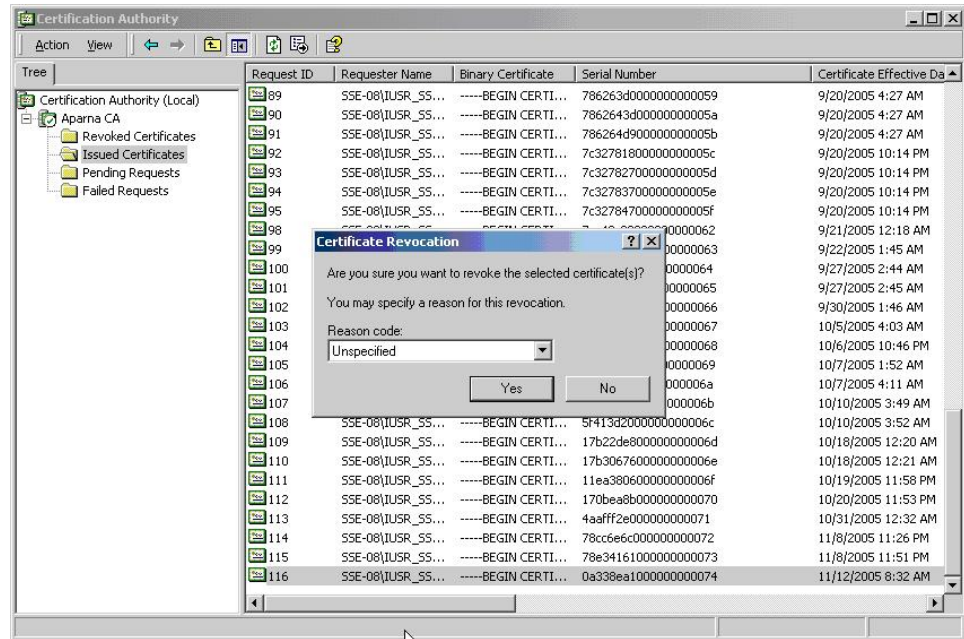
Step 4   Click the **Revoked Certificates** folder to list and verify the certificate revocation.
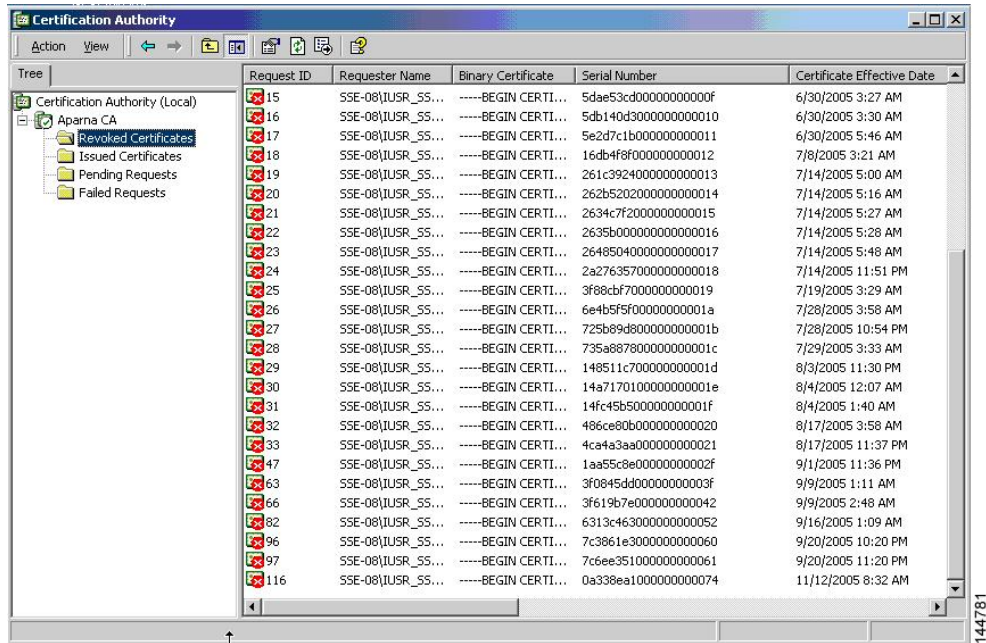
Generating and Publishing the CRL

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**.
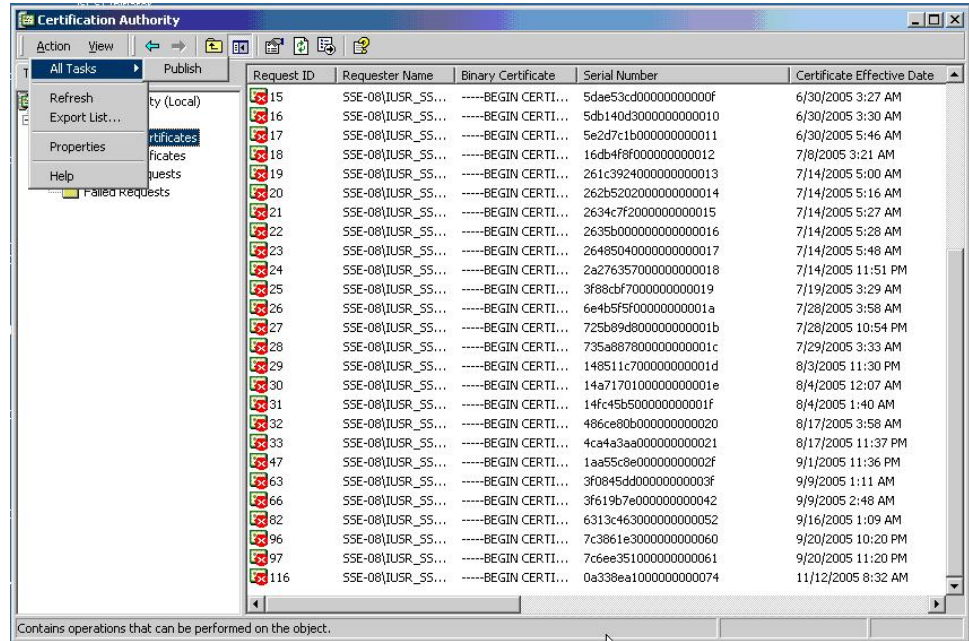
Step 2   In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.
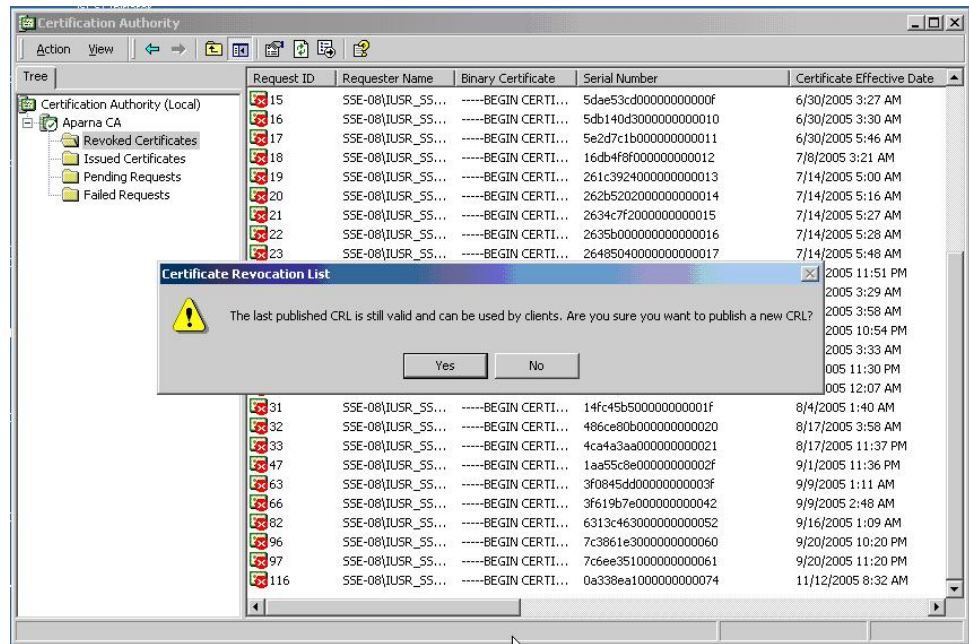
Downloading the CRL

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**.
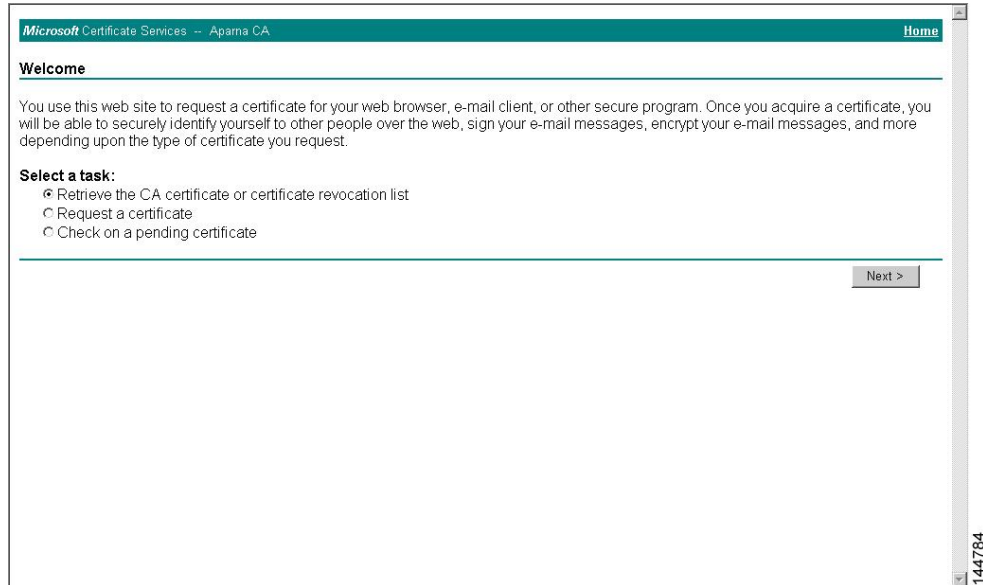
Step 2   Click **Download latest certificate revocation list**.
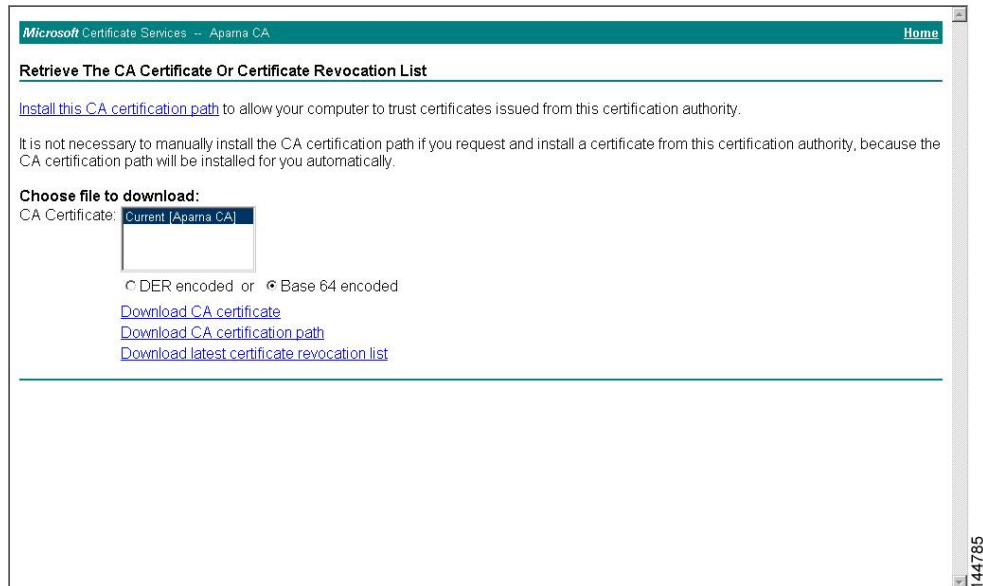
Step 3   In the File Download dialog box, click **Save**.
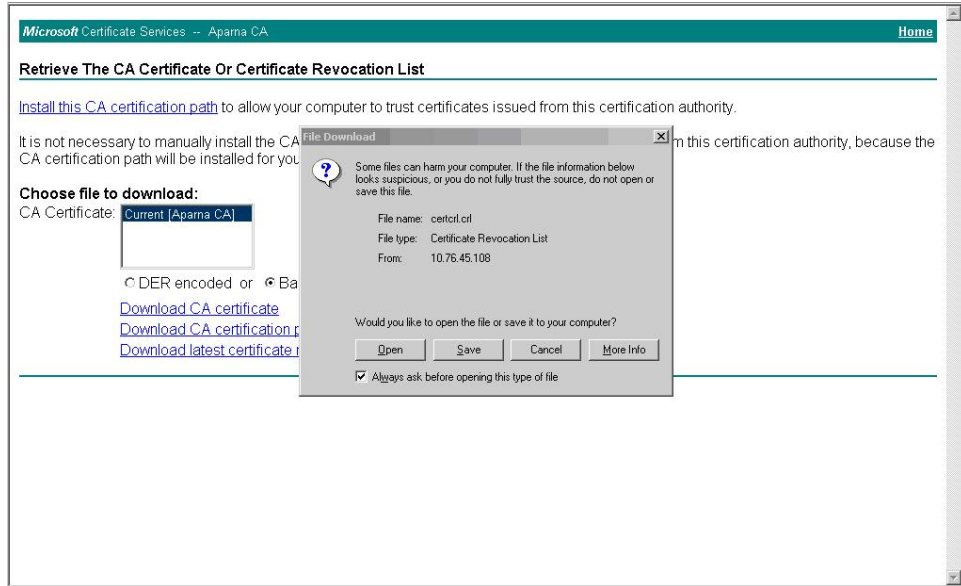
Step 4   In the Save As dialog box, enter the destination file name and click **Save**.

Step 5 Enter the Microsoft Windows **type** command to display the CRL.

Related Topics

Configuring Certificate Revocation Checking Methods

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 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: 5349AD460000000000 0A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C0000000000 0B
        Revocation Date: Jul  4 18:04:01 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE0000000000 0C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5D3FD52E0000000000 0D
        Revocation Date: Jun 29 22:07:25 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Key Compromise
Serial Number: 5DAB77130000000000 0E
        Revocation Date: Jul 14 00:33:56 2005 GMT
    Serial Number: 5DAE53CD0000000000 0F
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5DB140D3000000000010
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5E2D7C1B000000000011
        Revocation Date: Jul  6 21:12:10 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Cessation Of Operation
Serial Number: 16DB4F8F000000000012
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 261C3924000000000013
```

```
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 262B5202000000000014
        Revocation Date: Jul 14 00:33:10 2005 GMT
    Serial Number: 2634C7F2000000000015
        Revocation Date: Jul 14 00:32:45 2005 GMT
    Serial Number: 2635B000000000000016
        Revocation Date: Jul 14 00:31:51 2005 GMT
    Serial Number: 26485040000000000017
        Revocation Date: Jul 14 00:32:25 2005 GMT
    Serial Number: 2A2763570000000000018
Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 3F88CBF7000000000019
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 6E4B5F5F00000000001A
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 725B89D800000000001B
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 735A887800000000001C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 148511C700000000001D
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14A717010000000001E
        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
NX-OS Licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 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.	—

C H A P T E R 9

Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on NX-OS devices.

This chapter includes the following sections:

- About User Accounts and RBAC
- Licensing Requirements for User Accounts and RBAC
- Guidelines and Limitations for User Accounts and RBAC
- Default Settings for User Accounts and RBAC
- Enabling Password-Strength Checking
- Configuring User Accounts
- Configuring Roles
- About No Service Password-Recovery
- Enabling No Service Password-Recovery
- Verifying User Accounts and RBAC Configuration
- Configuration Examples for User Accounts and RBAC
- Additional References for User Accounts and RBAC=

About User Accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the 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:

- 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 contain dollar signs ($) or spaces anywhere in the password. Also, they cannot include these special characters at the beginning of the password: quotation marks (" or '), vertical bars (|), or right angle brackets (>).

✎

Note All printable ASCII characters are supported in the password string if they are enclosed in quotation marks.

If a password is trivial (such as a short, easy-to-decipher password), the 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.

Related Topics

Enabling Password-Strength Checking

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 NX-OS software provides the following user roles:

- network-admin—Complete read-and-write access to the entire NX-OS device

- network-operator or vdc-operator—Complete read access to the entire NX-OS device=

Note
- The CN93240YC-FX2 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 CN93240YC-FX2 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 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.

Licensing Requirements for User Accounts and RBAC

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	User accounts and RBAC require 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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 NX-OS device that has the same name as a remote user account on an AAA server, the 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 CN93240YC-FX2 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 IOS CLI, be aware that the NX-OS commands for this feature might differ from the IOS commands that you would use. |

Default Settings for User Accounts and RBAC

This table lists the default settings for user accounts and RBAC parameters.

Table 11: 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 NX-OS software does not check the strength of existing passwords. |

Procedure

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

	Command or Action	Purpose
Step 2	**password strength-check** **Example:** `switch(config)# password strength-check`	Enables password-strength checking. The default is enabled. You can disable password-strength checking by using the **no** form of this command.

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

Step 4	(Optional) **show password strength-check** **Example:** `switch# show password strength-check`	Displays the password-strength check configuration.

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

Related Topics

Characteristics of Strong Passwords

Configuring User Accounts

You can create a maximum of 256 user accounts on a 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 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.

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

Step 2	(Optional) **show role** **Example:** `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**] *password*] [**expire** *date*] [**role** *role-name*] **Example:** `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 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames. Usernames must begin with an alphanumeric character. The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the NX-OS device. **Note** If you create a user account with the encrypted password option, 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**} **Example:** `switch(config)# username NewUser ssh-cert-dn "/CN = NewUser, OU = Cisco Demo, O = Cisco, C = US" 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

	Command or Action	Purpose

	Example:
	`switch(config)# username jsmith`
	`ssh-cert-dn "/O = ABCcompany, OU = ABC1,`
	`emailAddress = jsmith@ABCcompany.com, L`
	`= Metropolis, ST = New York, C = US, CN`
	`= jsmith" rsa`
		are configured as emailAddress and ST, respectively.

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

Step 6	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the role configuration.

Step 7	(Optional) **copy running-config startup-config**
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

Configuring Roles

Creating User Roles and Rules

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.

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 NX-OS devices to which you want the configuration distributed.

Procedure

	Command or Action	Purpose

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

Step 2	**role name** *role-name*
	
	Example:
	`switch(config)# role name UserA`
	`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*
	
	Example:
	`switch(config-role)# rule 1 deny command`
	` clear users`
		Configures a command rule.
		
		The *command-string* argument can contain spaces and regular expressions. For example, interface ethernet includes all Ethernet interfaces.
		
		Repeat this command for as many rules as needed.

Step 4	**rule** *number* {**deny** | **permit**} {**read** | **read-write**}
	
	Example:
	`switch(config-role)# rule 2 deny`
	`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*
	
	Example:
	`switch(config-role)# rule 3 permit read`
	` feature router-bgp`
		Configures a read-only or read-and-write rule for a feature.
		
		Use the **show role feature** command to display a list of features.
		
		Repeat this command for as many rules as needed.

Step 6	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name*
	
	Example:
	`switch(config-role)# rule 4 deny`
	`read-write feature-group L3`
		Configures a read-only or read-and-write rule for a feature group.
		
		Use the **show role feature-group** command to display a list of feature groups.
		
		Repeat this command for as many rules as needed.

Step 7	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **oid** *snmp_oid_name*
	
	Example:
	`switch(config-role)# rule 5 deny`
	`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* **Example:** `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**
	
	Example:
	`switch(config-role)# exit`
	`switch(config)#`
		Exits role configuration mode.

Step 10	(Optional) **show role** **Example:** `switch(config)# show role`	Displays the user role configuration.
Step 11	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 12	(Optional) **role commit** **Example:** `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** **Example:** `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 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 NX-OS devices to which you want the configuration distributed.

Procedure

	Command or Action	**Purpose**

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

Step 2	**role feature-group name** *group-name*
	
	Example:
	`switch(config)# role feature-group name`
	`GroupA`
	`switch(config-role-featuregrp)#`
		Specifies a user role feature group and enters role feature group configuration mode.
		
		The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters.

Step 3	**feature** *feature-name*
	
	Example:
	`switch(config-role-featuregrp)# feature`
	`radius`
		Specifies a feature for the feature group.
		
		Repeat this command for as many features as needed.
		
		Note Use the **show role component** command to display a list of features.

Step 4	**exit**
	
	Example:
	`switch(config-role-featuregrp)# exit`
	`switch(config)#`
		Exits role feature group configuration mode.

Step 5	(Optional) **show role feature-group** **Example:** `switch(config)# show role feature-group`	Displays the role feature group configuration.
Step 6	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 7	(Optional) **role commit** **Example:** `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**
	
	Example:
	`switch(config)# copy running-config`
	`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 NX-OS devices to which you want the configuration distributed.

Procedure

	Command or Action	Purpose

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

Step 2	**role name** *role-name*
	
	Example:
	`switch(config)# role name UserA`
	`switch(config-role)#`
		Specifies a user role and enters role configuration mode.

Step 3	**interface policy deny**
	
	Example:
	`switch(config-role)# interface policy`
	`deny`
	`switch(config-role-interface)#`
		Enters role interface policy configuration mode.

Step 4	**permit interface** *interface-list*
	
	Example:
	`switch(config-role-interface)# permit`
	`interface ethernet 2/1-4`
		Specifies a list of interfaces that the role can access.
		
		Repeat this command for as many interfaces as needed.

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

Step 6	(Optional) **show role** **Example:** `switch(config-role)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config-role)# show role pending`	Displays the user role configuration pending for distribution.

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

Related Topics

[Creating User Roles and Rules](#)

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

Step 2	**role name** *role-name*
	
	Example:
	
	`switch(config)# role name UserA`
	`switch(config-role)#`
		Specifies a user role and enters role configuration mode.

Step 3	**vlan policy deny**
	
	Example:
	
	`switch(config-role)# vlan policy deny`
	`switch(config-role-vlan)#`
		Enters role VLAN policy configuration mode.

Step 4	**permit vlan** *vlan-list*
	
	Example:
	
	`switch(config-role-vlan)# permit vlan`
	`1-4`
		Specifies a range of VLANs that the role can access.
		
		Repeat this command for as many VLANs as needed.

Step 5	**exit** **Example:**	Exits role VLAN policy configuration mode.

	Command or Action	Purpose
	`switch(config-role-vlan)# exit` `switch(config-role)#`	
Step 6	(Optional) **show role** **Example:** `switch(config)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config-role)# show role pending`	Displays the user role configuration pending for distribution.
Step 8	(Optional) **role commit** **Example:** `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**
	
	Example:
	`switch(config-role)# copy running-config`
	` startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

Creating User Roles and Rules

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 NX-OS devices to which you want the configuration distributed.

Procedure

	Command or Action	Purpose

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

Step 2	**role name** *role-name*
	
	Example:
	`switch(config)# role name UserA`
	`switch(config-role)#`
		Specifies a user role and enters role configuration mode.

	Command or Action	Purpose

Step 3	**vrf policy deny**
	
	Example:
	
	`switch(config-role)# vrf policy deny`
	`switch(config-role-vrf)#`
		Enters role VRF policy configuration mode.

Step 4	**permit vrf** *vrf-name* **Example:** `switch(config-role-vrf)# permit vrf vrf1`	Specifies the VRF that the role can access. Repeat this command for as many VRFs as needed.

Step 5	**exit**
	
	Example:
	
	`switch(config-role-vrf)# exit`
	`switch(config-role)#`
		Exits role VRF policy configuration mode.

Step 6	(Optional) **show role** **Example:** `switch(config-role)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config-role)# show role pending`	Displays the user role configuration pending for distribution.
Step 8	(Optional) **role commit** **Example:** `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**
	
	Example:
	
	`switch(config-role)# copy running-config`
	` startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

Creating User Roles and Rules

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 CN93240YC-FX2 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, Inspur recommends that you save a copy of the system configuration file in a location away from the device.

Procedure

	Command or Action	Purpose

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

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

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

Step 4	**Reload**
	
	Example:
	`switch(config)# Reload`
	`This command will reboot the system.`
	`(y/n)?  [n] y`
	`2018 Jun 26 16:23:19 BAR %$ VDC-1 %$`
	`%PLATFORM-2-PFM_SYSTEM_RESET: Manual`
	`system restart from Command Line`
	`Interface`
	
	`CISCO SWITCH Ver 8.34`	

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

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
NX-OS Licensing	*NX-OS Licensing Guide*
VRF configuration	*CN93240YC-FX2 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 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on NX-OS devices. This chapter includes the following sections:

- About 802.1X
- Licensing Requirements for 802.1X
- Prerequisites for 802.1X
- 802.1X Guidelines and Limitations
- Default Settings for 802.1X
- Configuring 802.1X
- Verifying the 802.1X Configuration
- 802.1X Support for VXLAN EVPN
- Monitoring 802.1X
- Configuration Example for 802.1X
- Additional References for 802.1X=

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 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
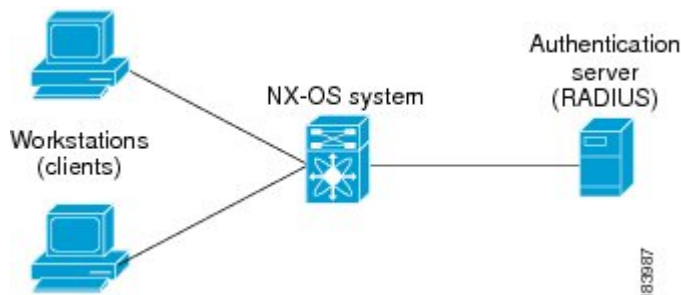
The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and NX-OS device services and responds to requests from the 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 NX-OS device regarding whether the supplicant is authorized to access the LAN and NX-OS device services. Because the 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 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 NX-OS device can only be an 802.1X authenticator.

Authentication Initiation and Message Exchange

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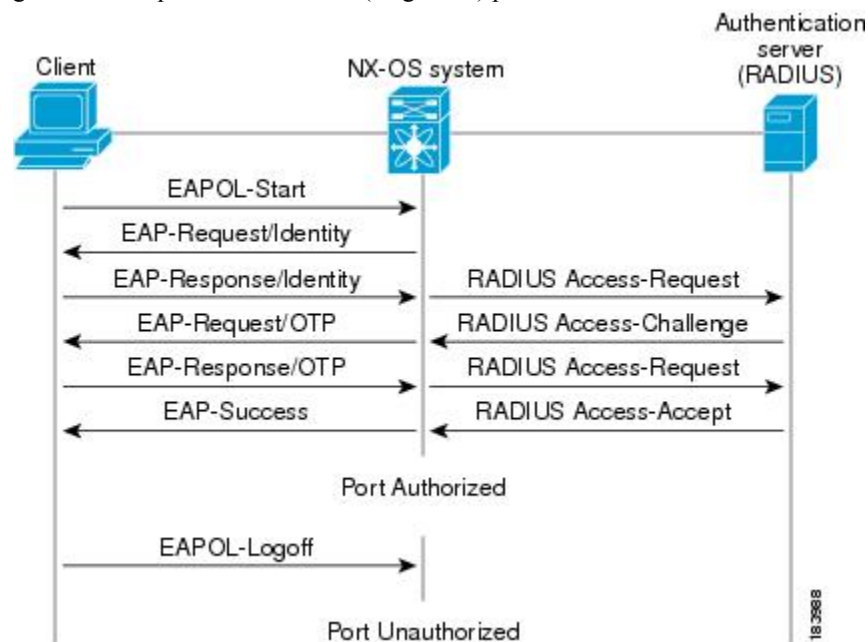
The user's secret pass-phrase never crosses the network at any time such as during authentication or during pass-phrase changes.

Authenticator PAE Status for Interfaces

When you enable 802.1X on an interface, the 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 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 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 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 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 NX-OS device waits for an Ethernet packet from the client. The 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 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 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 NX-OS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the NX-OS device does not unauthorize the client connected to the interface. When reauthentication occurs, the NX-OS device uses 802.1X authentication as the preferred reauthentication process.

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

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

MAC authentication bypass interacts with the following features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.

- Port security—You can configure 802.1X authentication and port security on the same Layer 2 ports.

- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

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

The CN93240YC-FX2 switches supports dynamic VLAN assignment. After the 802.1x authentication or MAB is completed; before bringing up the port, you may want to (as part of authorization) allow the peer/ host to be placed into a particular VLAN based as a result of the authentication. The RADIUS server

typically indicates the desired VLAN by including tunnel attributes within the Access-Accept message. This procedure of getting the VLAN an binding it to the port constitutes to Dynamic VLAN assignment.

VLAN Assignment from RADIUS

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

- Tunnel-type=VLAN(13)

- Tunnel-Medium-Type=802

- Tunnel-Private-Group-ID=VLANID

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

Single Host and Multiple Hosts Support

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

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the NX-OS device puts the port in the authorized state. When the endpoint device leaves the port, the 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 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 (NX-OS device) port. The authenticator detects the supplicant when the port link state changes to the up state. If a supplicant leaves or is replaced with another supplicant, the authenticator changes the port link state to down, and the port returns to the unauthorized state.

Licensing Requirements for 802.1X

The following table shows the licensing requirements for this feature:

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

Prerequisites for 802.1X

- CN Release 7.0(3)I7(1) software.=

802.1X Guidelines and Limitations

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

- When you upgrade the CN Series switch to NX-OS Release 9.2(1) using the (disruptive/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 NX-OS Release 9.2(1), multi-authentication mode is enabled on an 802.1X port. VLAN assignment occurs successfully for the first authenticated host. Subsequent authorized (based on user credentials) data hosts are considered successfully authenticated, provided either they have no VLAN assignment or have a VLAN assignment matching the first successfully authenticated host on the port. This ensures that all successfully authenticated hosts on a port are members of the same VLAN. Flexibility of VLAN assignment is only provided to the first authenticated host.

- Beginning with 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 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.

  - Dot1x with LACP port-channels.=

Note	Dot1x supports static port-channels.

Note	Disable 802.1X on VPC ports and all unsupported features.

- The NX-OS software supports 802.1X authentication only on physical ports.

- The NX-OS software does not support 802.1X authentication on port channels or subinterfaces.

- The NX-OS software supports 802.1X authentication on member ports of a port channel but not on the port channel itself.

- When the members are configured for 802.1X, NX-OS software does not support configuring single-host mode on port channel members. Only multi-host mode is supported on the member ports.

- Member ports with and without a 802.1X configuration can coexist in a port channel. However, you must ensure the identical 802.1X configuration on all the member ports in order for channeling to operate with 802.1X.

- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.

- On a 802.1X enabled port, the STP BPDUs are permitted only after a successful authentication. We recommend that you enable the 802.1X functionality only on the STP edge ports to avoid STP disputes.

- The NX-OS software supports 802.1X authentication only on Ethernet interfaces that are in a port channel, a trunk, or an access port.

- The NX-OS software does not work with the CTS or the MACsec features. Global "mac-learn disable" and dot1x feature are mutually exclusive and cannot be configured together.

- Dot1x is mutually exclusive with the IP Source Guard and URPF features and cannot be configured together. When you upgrade the CN Series switch to NX-OS Release 9.2(3), you must disable one of these features.

- The NX-OS software does not support single host mode on trunk interfaces or member interfaces in a port channel.

- The NX-OS software does not support MAC address authentication bypass on a port channel. The multi-host mode is the only supported mode on the port-channels.

- The NX-OS software does not support Dot1x on vPC ports and MCT.

- During a switch reload, Dot1x does not generate RADIUS accounting stops.

- The 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

- 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 dot1x enabled on the interface.

- Configuring mac learn disable with dot1x 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.=

Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

Table 12: Default 802.1X Parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (**force-authorized**) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet timeout period	60 seconds (number of seconds that the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant)
Retransmission timeout period	30 seconds (number of seconds that the Cisco NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)

Parameters	Default
Maximum retransmission number	2 times (number of times that the 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 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 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 NX-OS device before authenticating any supplicant devices.

Procedure

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

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

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

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.

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

Configuring AAA Authentication Methods for 802.1X

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose
		• *named-group* —Uses the global pool of RADIUS servers for authentication.

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

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 5	(Optional) **show radius-server group** [*group-name*] **Example:** `switch# show radius-server group rad2`	Displays the RADIUS server group configuration.

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

Controlling 802.1X Authentication on an Interface

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

Auto

Enables 802.1X authentication on the interface.

Force-authorized

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

Force-unauthorized

Disallows all traffic on the interface.

Before you begin

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

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

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

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

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

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

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 NX-OS software creates the authenticator PAE instance on the interface when you enable 802.1X on an interface.

Before you begin

Enable the 802.1X feature.

Procedure

	Command or Action	**Purpose**

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

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

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

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

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

Enabling Periodic Reauthentication for an Interface

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

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

Before you begin

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

Procedure

	Command or Action	**Purpose**

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

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

Step 3	**dot1x re-authentication**
	Example:
	`switch(config-if)# dot1x`
	`re-authentication`
		Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.

Step 4	(Optional) **dot1x timeout re-authperiod** *seconds*
	Example:
	`switch(config-if)# dot1x timeout`
	`re-authperiod 3300`
		Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535.
		
		Note This command affects the behavior of the NX-OS device only if you enable periodic reauthentication on the interface.

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

Step 6	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.

Step 7	(Optional) **copy running-config startup-config**
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Manually Reauthenticating Supplicants

You can manually reauthenticate the supplicants for the entire 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 NX-OS device.

Procedure

	Command or Action	**Purpose**
Step 1	**dot1x re-authenticate** [**interface** *slot*/*port*] **Example:** ``` switch# dot1x re-authenticate interface 2/1 ```	Reauthenticates the supplicants on the NX-OS device or on an interface.

Changing 802.1X Authentication Timers for an Interface

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

Quiet-period timer
> When the 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 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 NX-OS device with an EAP-response/identity frame. If the 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 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 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 NX-OS device.

Procedure

	Command or Action	**Purpose**

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

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

Step 3	(Optional) **dot1x timeout quiet-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`quiet-period 25`
		Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

Step 4	(Optional) **dot1x timeout ratelimit-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`ratelimit-period 10`
		Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.

Step 5	(Optional) **dot1x timeout server-timeout** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`server-timeout 60`
		Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.

Step 6	(Optional) **dot1x timeout supp-timeout** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`supp-timeout 20`
		Sets the number of seconds that the NX-OS device waits for the supplicant to respond to an EAP request frame before the NX-OS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Step 7	(Optional) **dot1x timeout tx-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`tx-period 40`
		Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

	Command or Action	Purpose

Step 8	(Optional) **dot1x timeout inactivity-period** *seconds*
	
	Example:
	`switch(config-if)# dot1x timeout`
	`inactivity-period 1800`
		Sets the number of seconds the switch can remain inactive. The recommended minimum value is1800 seconds.

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

Step 10	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays the 802.1X configuration.

Step 11	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Enabling MAC Authentication Bypass

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

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

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

	Command or Action	Purpose

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

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.

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

Enabling Single Host or Multiple Hosts Mode

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

Before you begin

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

Procedure

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

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

Step 3	**dot1x host-mode** {**multi-host** | **single-host**} **Example:** `switch(config-if)# `**`dot1x host-mode`** **`multi-host`**	Configures the host mode. The default is single-host. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.
Step 4	**dot1x host-mode multi-auth** **Example:**	Configures the multiple authentication mode. The port is authorized only on a successful authentication of either EAP or MAB or a

	Command or Action	Purpose
	switch(config-if)# **dot1x host-mode multi-auth**	combination of both. Failure to authenticate will restrict network access. authentication either EAP or MAB
Step 5	**exit** **Example:** switch(config-if)# **exit** switch(config)#	Exits configuration mode.
Step 6	(Optional) **show dot1x all** **Example:** switch# **show dot1x all**	Displays all 802.1X feature status and configuration information.
Step 7	(Optional) **copy running-config startup-config** **Example:** switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Disabling 802.1X Authentication on the NX-OS Device

You can disable 802.1X authentication on the NX-OS device. By default, the 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 NX-OS device. The 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 NX-OS software restores the configured port mode on the interfaces.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**no dot1x system-auth-control** **Example:**	Disables 802.1X authentication on the NX-OS device. The default is enabled.

	Command or Action	Purpose
	switch(config)# no dot1x system-auth-control	**Note** Use the **dot1x system-auth-control** command to enable 802.1X authentication on the NX-OS device.
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits configuration mode.
Step 4	(Optional) **show dot1x** **Example:** switch# show dot1x	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Disabling the 802.1X Feature

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

When you disable 802.1X, all related configurations are automatically discarded. The 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 *NX-OS System Management Configuration Guide* for your platform.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# **configure terminal** switch(config)#	Enters global configuration mode.
Step 2	**no feature dot1x** **Example:** switch(config)# **no feature dot1x**	Disables 802.1X. **Caution** Disabling the 802.1X feature removes all 802.1X configuration.
Step 3	**exit** **Example:**	Exits configuration mode.

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

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 NX-OS device.

Procedure

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

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

Step 3	**dot1x default** **Example:** switch(config-if)# dot1x default	Reverts to the 802.1X configuration default values for the interface.
Step 4	**exit** **Example:** switch(config-if)# exit switch(config)#	Exits configuration mode.
Step 5	(Optional) **show dot1x all** **Example:** switch(config)# show dot1x all	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Setting the Maximum Authenticator-to-Supplicant Frame for an Interface

You can set the maximum number of times that the 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 NX-OS device.

Procedure

	Command or Action	**Purpose**

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

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

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

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

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

Enabling RADIUS Accounting for 802.1X Authentication

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

Before you begin

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

Procedure

	Command or Action	Purpose

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

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

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

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X configuration.

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

Configuring AAA Accounting Methods for 802.1X

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

Before you begin

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

Procedure

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

	Command or Action	**Purpose**
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 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 X-OS device.

Procedure

	Command or Action	**Purpose**

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

	Command or Action	Purpose

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

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

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

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

Verifying the 802.1X Configuration

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

Command	Purpose
show dot1x	Displays the 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.
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 *NX-OS Security Command Reference* for your platform.

802.1X Support for VXLAN EVPN

This section describes how to configure configures 802.1X for VXLAN EVPN.

Guidelines and Limitations for 802.1X Support for VXLAN EVPN

The following are the guidelines and limitations for 802.1X support for VXLAN EVPN:

- Port channel interfaces or the member ports of the port channel are not supported.

- vPC ports are not supported.

- The current support of the feature uses regular and dynamic EVPN updates on the BGP-EVPN control plane for 802.1X secure MAC updates. As a result, we cannot prevent the move across EVPN even if the global policy is "dot1x mac-move deny".

- Ensure that the "dot1x mac-move" policy is configured the same across the fabric. There is no configuration validation across the nodes, hence it could lead to unexpected behavior if the configuration policy is not in sync.

- The local to remote MAC moves behavior for the deny and permit modes is permitted. Therefore, the MAC move is permitted even if the deny mode is enabled.

- Ensure that the dot1x and the port-security ports use different VLANs. The same VLAN cannot be assigned to both ports.

- Dot1x is not VLAN aware and hence having the same MAC in two different VLANs is not possible. Depending on the mac-move mode that is selected, either the MAC is moved to a new VLAN or it is denied.

- You cannot configure static and secure MAC together.

- RADIUS change of Authorization is supported for VXLAN EVPN.

- The recommended re-authentication time interval for a scale setup is the default value, which is 3600 seconds.=

Configuring 802.1X Support for VXLAN EVPN

This procedure configures 802.1X for VXLAN EVPN.

Procedure

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

	Command or Action	Purpose
Step 2	**feature dot1x** **Example:** switch(config)# **feature dot1x**	Enables the 802.1X feature. The default is disabled.
Step 3	**dot1x mac-move {permit | deny}** **Example:** switch(config)# **dot1x mac-move permit**	The **deny** parameters denies MAC moves. The **permit** parameter permits MAC moves.

Step 4	(Optional) **show running-config dot1x all**
	
	Example:
	
	switch(config)# **show running-config dot1x all**
	
	!Command: show running-config dot1x all
	!No configuration change since last restart
	!Time: Thu Sep 20 10:22:58 2018
	
	version 9.2(2) Bios:version 07.64
	feature dot1x
	
	dot1x system-auth-control
	dot1x mac-move deny
	
	interface Ethernet1/1
	  dot1x host-mode multi-auth
	  dot1x pae authenticator
	  dot1x port-control auto
	  no dot1x re-authentication
	  dot1x max-req 1
	  dot1x max-reauth-req 2
	  dot1x timeout quiet-period 60
	  dot1x timeout re-authperiod 3600
	  dot1x timeout tx-period 1
	  dot1x timeout server-timeout 30
	  dot1x timeout ratelimit-period 0
	  dot1x timeout supp-timeout 30
	  dot1x timeout inactivity-period 0
	  dot1x mac-auth-bypass
	
	interface Ethernet1/33
	  dot1x host-mode multi-auth
	  dot1x pae authenticator
	  dot1x port-control auto
	  no dot1x re-authentication
	  dot1x max-req 1
	  dot1x max-reauth-req 2
	  dot1x timeout quiet-period 60
	  dot1x timeout re-authperiod 3600
	  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
		Displays the 802.1X configuration.

Verifying the 802.1X Support for VXLAN EVPN

To display the 802.1X Support for VXLAN ENPN configuration information, enter one of the following commands:

Command	Purpose
show running-config dot1x all	Displays dot1x running configuration.
show dot1x all summary	Displays the interface status.
show dot1x	Displays the default settings.
show dot1x all	Displays additional interface detail.

Example of show running-config dot1x all command

```
switch# show running-config dot1x all
!Command: show running-config dot1x all
!No configuration change since last restart
!Time: Thu Sep 20 10:22:58 2018

version 9.2(2) Bios:version 07.64
feature dot1x

dot1x system-auth-control
dot1x mac-move deny


interface Ethernet1/1
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
  dot1x max-req 1
  dot1x max-reauth-req 2
  dot1x timeout quiet-period 60
  dot1x timeout re-authperiod 3600
  dot1x timeout tx-period 1
  dot1x timeout server-timeout 30
  dot1x timeout ratelimit-period 0
  dot1x timeout supp-timeout 30
  dot1x timeout inactivity-period 0
  dot1x mac-auth-bypass

interface Ethernet1/33
  dot1x host-mode multi-auth
  dot1x pae authenticator
  dot1x port-control auto
  no dot1x re-authentication
  dot1x max-req 1
  dot1x max-reauth-req 2
  dot1x timeout quiet-period 60
  dot1x timeout re-authperiod 3600
  dot1x timeout tx-period 1
  dot1x timeout server-timeout 30
  dot1x timeout ratelimit-period 0
  dot1x timeout supp-timeout 30
  dot1x timeout inactivity-period 0
  dot1x mac-auth-bypass
```

Example of the show dot1x all summary command

```
switch# show dot1x all summary

           Interface     PAE            Client          Status
------------------------------------------------------------------
          Ethernet1/1    AUTH           none    UNAUTHORIZED

           Interface     PAE            Client          Status
------------------------------------------------------------------
          Ethernet1/33   AUTH  00:16:5A:4C:00:07      AUTHORIZED
                               00:16:5A:4C:00:06      AUTHORIZED
                               00:16:5A:4C:00:05      AUTHORIZED
                               00:16:5A:4C:00:04      AUTHORIZED
switch#
switch# show mac address-table vlan 10
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
     VLAN      MAC Address      Type      age      Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10    0016.5a4c.0004   secure    -         T      F    Eth1/33
*   10    0016.5a4c.0005   secure    -         T      F    Eth1/33
*   10    0016.5a4c.0006   secure    -         T      F    Eth1/33
*   10    0016.5a4c.0007   secure    -         T      F    Eth1/33

switch#
switch# show mac address-table vlan 10 (VPC-PEER)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
     VLAN      MAC Address      Type      age      Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10    0016.5a4c.0004   secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0005   secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0006   secure    -         T      F    vPC Peer-Link
*   10    0016.5a4c.0007   secure    -         T      F    vPC Peer-Link

switch#
switch# show mac address-table vlan 10 (RVTEP)
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
     VLAN      MAC Address      Type      age      Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C   10    0016.5a4c.0004   dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0005   dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0006   dynamic  0         F      F    nve1(67.67.67.67)
C   10    0016.5a4c.0007   dynamic  0         F      F    nve1(67.67.67.67)
```

Example of the show dot1x command

```
switch# show dot1x
        Sysauthcontrol Enabled
  Dot1x Protocol Version 2
           Mac-Move Deny
```

Example of the show dot1x all command

```
switch# show dot1x  all
           Sysauthcontrol Enabled
   Dot1x Protocol Version 2
                 Mac-Move Deny

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

Dot1x Info for Ethernet1/33
----------------------------------
                      PAE = AUTHENTICATOR
              PortControl = AUTO
                 HostMode = MULTI AUTH
          ReAuthentication = Disabled
              QuietPeriod = 60
            ServerTimeout = 30
              SuppTimeout = 30
             ReAuthPeriod = 3600 (Locally configured)
               ReAuthMax = 2
                  MaxReq = 1
                 TxPeriod = 1
          RateLimitPeriod = 0
          InactivityPeriod = 0
           Mac-Auth-Bypass = Enabled
```

Monitoring 802.1X

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

Before you begin

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

Procedure

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

Configuration Example for 802.1X

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

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

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

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

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

Additional References for 802.1X

This section includes additional information related to implementing 802.1X.

Related Documents

Related Topic	Document Title
NX-OS Licensing	*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*

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on 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 13: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	• Layer 2 interfaces • Layer 2 Ethernet port-channel interfaces When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	• IPv4 ACLs • IPv6 ACLs • MAC ACLs • MAC ACLs with UDF-based match.=

Application	Supported Interfaces	Types of ACLs Supported
Router ACL	• VLAN interfaces • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Management interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface.	• IPv4 ACLs • IPv6 ACLs **Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs
VTY ACL	• VTYs	• IPv4 ACLs • IPv6 ACLs

Related Topics

About VLAN ACLs

About MAC ACLs

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.
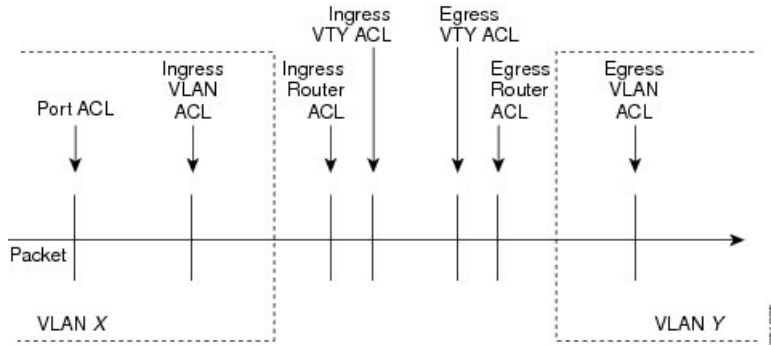
Figure 8: ACLs and Packet Flow

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.
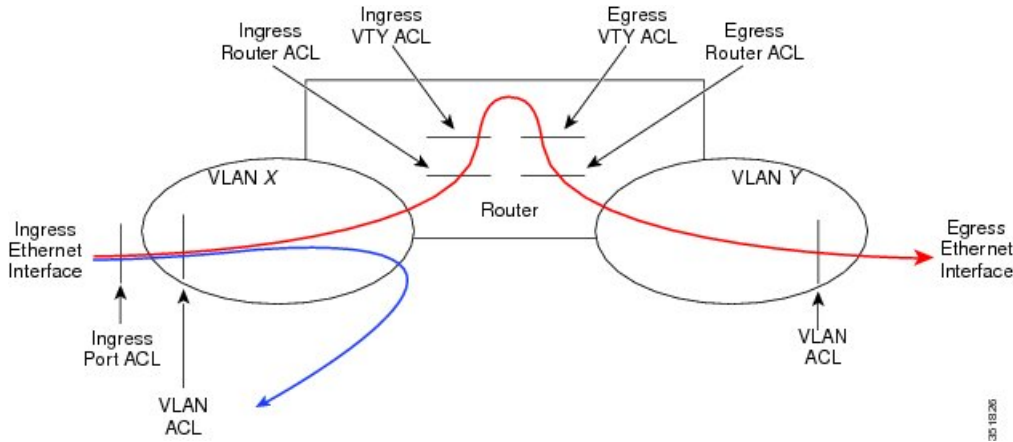
About Rules

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 nd-na, nd-ns, router-advertisement, and router-solicitation packets will not be permitted as the implicit permit rules on IPv6 ACL. You must add the following rules explicitly to allow them:

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

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

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

Additional Filtering Options

You can identify traffic by using additional options. 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)=

• Established TCP connections

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, 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.

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

IPv4 ACL Logging

The IPv4 ACL logging feature monitors IPv4 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).

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. 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.

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

Monitoring and Clearing IP ACL Statistics

Implicit Rules for IP and MAC ACLs

Atomic ACL Updates

By default, when a supervisor module of a CN93240YC-FX2 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.

You can create IPv6, port ACLs, VLAN ACLs, and router ACLs, and you can match IPv6 and MAC addresses for QoS. However, 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.

ACL TCAM region sizes have the following guidelines and limitations:

- To enable RACL or PACL on existing TCAM regions, you must carve the TCAM region beyond 12, 288.

- 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.

- 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
```

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 14: Features per ACL TCAM Region

Feature Name	Region Name
Port ACL	ifacl: For IPv4 port ACLs 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
VACL	vacl: For IPv4 packets ipv6-vacl: For IPv6 packets mac-vacl: For non-IP packets

Feature Name	Region Name
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 mac-vqos or ns-mac-vqos: For classifying non-IP packets
RACL	e-racl: For egress IPv4 RACLs e-ipv6-racl: For egress IPv6 RACLs ing-racl: For IPv4 RACLs 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 ipv6-l3qos or ns-ipv6-l3qos: For classifying IPv6 packets

Feature Name	Region Name
SPAN filters	ifacl: For filtering IPv4 traffic on Layer 2 (switch port) source interfaces.
	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.
	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.
SVI counters **Note** This region enables the packet counters for Layer 3 SVI interfaces.	svi

Feature Name	Region Name
BFD, DHCP relay, or DHCPv6 relay	redirect
CoPP	copp **Note** The region size cannot be 0.
System-managed ACLs	system **Note** The region size cannot be changed.
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.	vpc-convergence **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

Related Topics

Configuring ACL TCAM Region Sizes

Configuring TCAM Carving

Maximum Label Sizes Supported for ACL Types

NX-OS switches support the following label sizes for the corresponding ACL

types: **Table 15: 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	30	IF
RACL/VACL/L3-VLAN QoS	Egress	254	BD
L2 QoS	Egress	31	IF

Licensing Requirements for IP ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	No license is required to use IP ACLs. 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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:

• 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 *CN93240YC-FX2 NX-OS System Management Configuration Guide*.

• 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.

- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:

    - Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.

    - IPv4 packets that have IP options (additional IP packet header fields following the destination address field).

    - IPv6 packets that have extended IPv6 header fields.=

    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 *CN93240YC-FX2 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.

- 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.=

- IPv6 ACL logging is not supported.

- 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.

- 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 need to be expanded to multiple entries in the egress TCAM. Make sure to consider this limitation for egress TCAM space planning.

- 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 physical layer 3 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.=

- In NX-OS Release 9.2(3), 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=

Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 16: 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

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

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

	Command or Action	**Purpose**
	`switch(config)# ip access-list acl-01` `switch(config-acl)#`	
Step 3	(Optional) **fragments** {**permit-all** | **deny-all**} **Example:** `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*} **Example:** `switch(config-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For 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 destination IPv4 or IPv6 wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for CN93240YC-FX2 switch.
Step 5	(Optional) **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 6	**hardware profile acl-stats module** *xx* **Example:** `switch(config-acl)# hardware profile acl-stats module 10`	Enables counters for the ACL TCAM entries on both, the internal and external TCAM.
Step 7	**reload module** *xx*	Reloads the switch.

	Command or Action	Purpose
	Example: `switch(config)# reload module 10`	
Step 8	**ignore routeable** **Example:** `switch(config)# ignore routeable`	
Step 9	(Optional) Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

	Command or Action	**Purpose**

Step 2	Enter one of the following commands:
	
	    • **ip access-list** *name*
	    • **ipv6 access-list** *name*
	
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Enters IP ACL configuration mode for the ACL that you specify by name.

Step 3	(Optional) [*sequence-number*] {**permit** | **deny**} *protocol source destination*
	
	Example:
	`switch(config-acl)# 100 permit ip`
	`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.
		
		The **permit** and **deny** commands support many ways of identifying traffic.

Step 4	(Optional) [**no**] **fragments** {**permit-all** | **deny-all**} **Example:** `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. The **no** option removes fragment-handling optimization.
Step 5	(Optional) **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} **Example:** `switch(config-acl)# no 80`	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic.
Step 6	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.

Step 7	(Optional) Enter one of the following commands:
	
	    • **show ip access-lists** *name*
	    • **show ipv6 access-lists** *name*
	
	Example:
	`switch(config-acl)# show ip access-lists`
	`acl-01`
		Displays the IP ACL configuration.

	Command or Action	Purpose
Step 8	(Optional) **copy running-config startup-config** **Example:** `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

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

Step 2	{**ip** | **ipv6**} **access-list** *name* **Example:** `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*] **Example:** `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**
	
	Example:
	`switch(config-ip-acl)# exit`
	`switch(config)#`
		Exits IP access list configuration mode.

	Command or Action	Purpose

Step 5	**line vty**
	
	Example:
	
	`switch(config)# line vty`
	`switch(config-line)#`
		Specifies the virtual terminal and enters line configuration mode.

Step 6	**{ip | ipv6} access-class** *name* **{in | out}**
	
	Example:
	
	`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** **Example:** `switch# show ip access-lists`	Displays the configured ACLs, including any VTY ACLs.

Step 8	(Optional) **copy running-config startup-config**
	
	Example:
	
	`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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence {ip | ipv6} access-list** *name starting-sequence-number increment*
	
	Example:
	
	`switch(config)# resequence access-list`
	`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*

	Command or Action	Purpose
		argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 2	Enter one of the following commands: • **no ip access-list** *name* • **no ipv6 access-list** *name* **Example:** `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: • **show ip access-lists** *name* **summary** • **show ipv6 access-lists** *name* **summary** **Example:**	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.

	Command or Action	Purpose
	switch(config)# show ip access-lists acl-01 summary	
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring 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.

Note

- Once you apply a template (using Using Templates to Configure ACL TCAM Region Sizes, on page 285), 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 CN93240YC-FX2 switches.

- For information on configuring QoS TCAM carving, see the *CN93240YC-FX2 NX-OS Quality of Service Configuration Guide*.=

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **hardware access-list tcam region** *region tcam-size* **Example:** switch(config)# hardware access-list tcam region mpls 256	Changes the ACL TCAM region size. These are the available regions: - **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.

Command or Action	Purpose
	• **arp-ether**—Configures the size of the ARP/Layer 2 Ethertype TCAM region.
	• **copp**—Configures the size of the CoPP TCAM region.
	• **e-flow**—Configures the size of the egress flow counters TCAM region.
	• **egr-copp**—Configures the size of the egress CoPP TCAM region.
	• **egr-racl**—Configures the size of the egress IPv4 or IPv6 router ACL (RACL) TCAM region (Cisco Nexus 9200 switches only).
	• **egr-sup**—Configures the size of the egress supervisor TCAM region (Cisco Nexus 9200 switches only).
	• **e-ipv6-qos**—Configures the size of the IPv6 egress QoS TCAM region.
	• **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.

Command or Action	Purpose
	• **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 (Cisco Nexus 3232C and 3264Q switches only).
	• **Note** You can attach user-defined fields (UDFs) to the ing-ifacl TCAM region to configure UDF-based IPv4 or IPv6 port ACLs. For more information and configuration instructions, see Configuring UDF-Based Port ACLs.=

Command or Action	Purpose
	• **ing-l3-vlan-qos**—Configures the size of the ingress Layer 3, VLAN, and SVI QoS TCAM region (Cisco Nexus 9200 switches only).
	• **ing-netflow**—Configures the size of the NetFlow 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.
	• **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.=

Command or Action	Purpose
	• **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.=

Command or Action	Purpose
	• **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.
	• **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.=
	Note This command is supported only if the TP_SERVICES_PKG license is installed.

Command or Action	Purpose
	• **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 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.

	Command or Action	Purpose
		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 *CN93240YC-FX2 NX-OS System Management Configuration Guide*.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	(Optional) **show hardware access-list tcam region** **Example:** `switch(config)# show hardware access-list tcam region`	Displays the TCAM sizes that will be applicable on the next reload of the device.
Step 5	**reload** **Example:** `switch(config)# reload`	Reloads the device. **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 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
```

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
```

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 CN93240YC-FX2 switches, you can use this procedure or the "Configuring ACL TCAM Region Sizes" procedure to configure ACL TCAM region sizes.

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 *CN93240YC-FX2 NX-OS Quality of Service Configuration Guide*.=

Procedure

	Command or Action	**Purpose**

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

Step 2	[**no**] **hardware profile tcam resource template** *template-name* **ref-template** {**nfe** | **nfe2** | {**l2-l3** | **l3**}}
	
	Example:
	
	`switch(config)# hardware profile tcam`
	`resource template SR_MPLS_CARVE`
	`ref-template nfe2`
	`switch(config-tcam-temp)#`
		Creates a template for configuring ACL TCAM region sizes.

Step 3	(Optional) *region tcam-size* **Example:** `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**
	
	Example:
	
	`switch(config-tcam-temp)# exit`
	`switch(config#)`
		Exits the TCAM template configuration mode.

Step 5	[**no**] **hardware profile tcam resource service-template** *template-name*
	
	Example:
	
	`switch(config)# hardware profile tcam`
	`resource service-template SR_MPLS_CARVE`
		Applies the custom template to all line cards and fabric modules.

Step 6	(Optional) **show hardware access-list tcam template** {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*} **Example:**	Displays the configuration for all TCAM templates or for a specific template.

	Command or Action	Purpose
	switch(config)# show hardware access-list tcam template SR_MPLS_CARVE	
Step 7	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.
Step 8	**reload** **Example:** switch(config)# reload	Reloads the device. **Note** The configuration is effective only after you enter **copy running-config startup-config** + **reload**.

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 30: 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[1]
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

[1] 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 31: 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
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 32: 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

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.

Related Topics

Configuring ACL TCAM Region Sizes

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

Step 2	Enter one of the following commands:
	
	 • **interface ethernet** *slot*/*port*[**.** *number*]
	 • **interface port-channel** *channel-number*
	 • **interface vlan** *vlan-id*
	 • **interface mgmt** *port*
	
	 Example:
	 ```switch(config)# interface ethernet 2/3```
	 ```switch(config-if)#```
		Enters configuration mode for the interface type that you specified.

Step 3	Enter one of the following commands: • **ip access-group** *access-list* {**in** | **out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction

Command or Action	Purpose

• **ipv6 traffic-filter** *access-list* {**in** | **out**}
 	
Example:
`switch(config-if)# ip access-group acl1`
`  in`
	specified. You can apply one router ACL per direction.

Step 4 (Optional) **show running-config aclmgr**
 	
Example:
`switch(config-if)# show running-config`
`aclmgr`
	Displays the ACL configuration.

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

Related Topics

Creating an IP ACL

Applying an IP ACL as a Port ACL

You can apply an IPv4 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**
 	
Example:
`switch# configure terminal`
`switch(config)#`
	Enters global configuration mode.

Step 2 Enter one of the following commands: • **interface ethernet** *slot/port* • **interface port-channel** *channel-number* **Example:**	Enters configuration mode for the interface type that you specified.

	Command or Action	Purpose
	`switch(config)# interface ethernet 2/3` `switch(config-if)#`	

Step 3	Enter one of the following commands:
	
	    • **ip port access-group** *access-list* **in**
	    • **ipv6 port traffic-filter** *access-list* **in**
	
	Example:
	`switch(config-if)# ip port access-group`
	`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**
	
	Example:
	`switch(config-if)# show running-config`
	`aclmgr`
		Displays the ACL configuration.

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

Related Topics

Creating an IP ACL

Enabling or Disabling MAC Packet Classification

Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

Related Topics

Configuring VACLs

Configuring IPv4 ACL Logging

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

Procedure

	Command or Action	Purpose

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

	Command or Action	**Purpose**

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

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

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

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

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

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

Step 8	**logging ip access-list cache interval** *interval*
	
	Example:
	`switch(config)# logging ip access-list`
	`cache interval 490`
		Configures the log-update interval (in seconds) for the ACL logging process. The default value is 300 seconds. The range is from 5 to 86400 seconds.

Step 9	**logging ip access-list cache entries** *number-of-flows*
	
	Example:
	`switch(config)# logging ip access-list`
	`cache entries 8001`
		Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576.

	Command or Action	Purpose
Step 10	**logging ip access-list cache threshold** *threshold* **Example:** `switch(config)# logging ip access-list cache threshold 490`	If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message.
Step 11	**logging ip access-list detailed** **Example:** `switch(config)# logging ip access-list detailed`	Enables the 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* **Example:** `switch(config)# hardware rate-limiter access-list-log 200`	Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000.
Step 13	**acllog match-log-level** *severity-level* **Example:** `switch(config)# acllog match-log-level 5`	Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging).
Step 14	(Optional) **show logging ip access-list cache** [**detail**] **Example:** `switch(config)# show logging ip access-list cache`	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces, and so on. 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.

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

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

Step 3	[*sequence-number*] **permit** *protocol source destination* **http-method** *method* [**tcp-option-length** *length*] [**redirect** *interface*] **Example:** `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. The following HTTP methods are supported: - connect—Matches HTTP packets with the CONNECT method [0x434f4e4e] - delete—Matches HTTP packets with the DELETE method [0x44454c45] - get—Matches HTTP packets with the GET method [0x47455420] - 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

	Command or Action	Purpose
		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*
	
	Example:
	`switch(config-acl)# show ip access-lists`
	`acl-01`
		Displays the IP ACL configuration.

Step 5	(Optional) **show run interface** *interface slot/port*
	
	Example:
	`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
```

Verifying the IP ACL Configuration

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

Command	Purpose
show hardware access-list tcam region	Displays the TCAM sizes that will be applicable on the next reload of the device.

Command	Purpose
show hardware access-list tcam template {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}	Displays the configuration for all TCAM templates or for a specific template.
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. 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.
show logging ip access-list status	Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value.
show running-config acllog	Displays the ACL log running configuration.

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config acllog	Displays the ACL log startup configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Monitoring and Clearing IP ACL Statistics

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.

Command	Purpose
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
```

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 section for more information.

Note Beginning with 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 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 *CN93240YC-FX2 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 CN 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**object-group ip address** *name*
	
	Example:
	
	`switch(config)# object-group ip address`
	`ipv4-addr-group-13`
	`switch(config-ipaddr-ogroup)#`
		Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.

Step 3	Enter one of the following commands:
	
	• [*sequence-number*] **host** *IPv4-address*
	• [*sequence-number*] *IPv4-address/prefix-len*
	• [*sequence-number*] *IPv4-address network-wildcard*
	
	Example:
	
	`switch(config-ipaddr-ogroup)# host`
	`10.99.32.6`
		Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host, or omit the **host** command to specify a network of hosts.
		
		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:
	
	• **no** [*sequence-number*]
	• **no host** *IPv4-address*
	• **no** *IPv4-address/prefix-len*
	• **no** *IPv4-address network-wildcard*
	
	Example:
	
	`switch(config-ipaddr-ogroup)# no host`
	`10.99.32.6`
		Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.

Step 5	(Optional) **show object-group** *name*
	
	Example:
	
	`switch(config-ipaddr-ogroup)# show`
	`object-group ipv4-addr-group-13`
		Displays the object group configuration.

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

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

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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**object-group ipv6 address** *name* **Example:** switch(config)# object-group ipv6 address ipv6-addr-group-A7 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: • [*sequence-number*] **host** *IPv6-address* • [*sequence-number*] *IPv6-address/prefix-len* • [*sequence-number*] *IPv6-address network-wildcard* **Example:** switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1 **Example:** 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. 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 CN93240YC-FX2 switch.
Step 4	Enter one of the following commands: • **no** *sequence-number* • **no host** *IPv6-address* • **no** *IPv6-address/prefix-len* • **no** *IPv6-address network-wildcard* **Example:** switch(config-ipv6addr-ogroup)# no host 2001:db8:0:3ab0::1	Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:**	Displays the object group configuration.

	Command or Action	Purpose
	switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7	
Step 6	(Optional) **copy running-config startup-config** **Example:** switch(config-ipv6addr-ogroup)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Creating and Changing a Protocol Port Object Group

You can create and change a protocol port object group.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**object-group ip port** *name* **Example:** switch(config)# object-group ip port NYC-datacenter-ports 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*] **Example:** switch(config-port-ogroup)# eq 80	Creates an entry in the object group. For each entry that you want to create, use one of the following operator commands: • **eq**—Matches only the port number that you specify. • **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.

	Command or Action	Purpose
		Note The **range** command is the only operator command that requires two *port-number* arguments.
Step 4	**no** {*sequence-number*	*operator port-number* [*port-number*]} **Example:** `switch(config-port-ogroup)# no eq 80`	Removes an entry from the object group. For each entry that you want to remove, use the **no** form of the applicable operator command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing an Object Group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

Procedure

	Command or Action	Purpose

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

Step 2	**no object-group** {**ip address**	**ipv6 address**	**ip port**} *name* **Example:** `switch(config)# no object-group ip address ipv4-addr-group-A7`	Removes the specified object group.
Step 3	(Optional) **show object-group** **Example:** `switch(config)# show object-group`	Displays all object groups. The removed object group should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

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 *CN93240YC-FX2 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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**time-range** *name* **Example:** switch(config)# time-range workday-daytime switch(config-time-range)#	Creates the time range and enters time-range configuration mode.

	Command or Action	Purpose
Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `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* **Example:** `switch(config-time-range)# periodic 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: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `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* **Example:** `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) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**time-range** *name*
	
	Example:
	`switch(config)# time-range`
	`workday-daytime`
	`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*
	
	Example:
	`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*
	
	Example:
	`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:
		
		• **daily** —All days of the week.
		
		• **weekdays** —Monday through Friday.
		
		• **weekend** —Saturday through Sunday.

Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]
	
	Example:
	`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*
	
	Example:
	`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. . .*}	Removes the specified rule from the time range.

	Command or Action	Purpose
	Example: switch(config-time-range)# no 80	
Step 8	(Optional) **show time-range** *name* **Example:** switch(config-time-range)# show time-range workday-daytime	Displays the time-range configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** switch(config-time-range)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in a Time Range

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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**no time-range** *name* **Example:** switch(config)# no time-range daily-workhours	Removes the time range that you specified by name.
Step 3	(Optional) **show time-range** **Example:** switch(config-time-range)# show time-range	Displays the configuration for all time ranges. The removed time range should not appear.

	Command or Action	Purpose

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence time-range** *name*
	starting-sequence-number increment
	
	Example:
	`switch(config)# resequence time-range`
	`daily-workhours 100 10`
	`switch(config)#`
		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*
	
	Example:
	`switch(config)# show time-range`
	`daily-workhours`
		Displays the time-range configuration.

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

Verifying the 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

Inspur-Cisco Networking
Technology Co.,Ltd.

Configuring MAC ACLs

This chapter describes how to configure MAC access lists (ACLs) on 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.

Note   MAC packet classification is not supported in NX-OS Release 9.3(3).

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. - You cannot apply an IP port ACL on the interface.

MAC Packet Classification State	Effect on Interface
Disabled	• A MAC ACL that is on the interface applies only to non-IP traffic entering the interface. • You can apply an IP port ACL on the interface

Licensing Requirements for MAC ACLs

This table shows the licensing requirements for this feature.

Product	License Requirement
NX-OS	MAC ACLs require 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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.

- 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.=

Default Settings for MAC ACLs

This table lists the default settings for MAC ACL parameters.

Table 33: Default MAC ACLs Parameters

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

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it.

Procedure

	Command or Action	Purpose

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

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

Step 3	{**permit** | **deny**} *source destination-protocol*
	
	Example:
	
	`switch(config-mac-acl)# 100 permit mac`
	`00c0.4f00.0000 0000.00ff.ffff any 0x0806`
		Creates a rule in the MAC ACL.
		
		The **permit** and **deny** commands support many ways of identifying traffic.

Step 4	(Optional) **statistics per-entry**
	
	Example:
	
	`switch(config-mac-acl)# statistics`
	`per-entry`
		Specifies that the device maintains global statistics for packets that match the rules in the ACL.

	Command or Action	**Purpose**
Step 5	(Optional) **show mac access-lists** *name* **Example:** `switch(config-mac-acl)# show mac access-lists acl-mac-01`	Displays the MAC ACL configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-mac-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an Interface MAC Address and Limit

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

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

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

Procedure

	Command or Action	**Purpose**

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

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

	Command or Action	**Purpose**

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

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

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

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

Example

The following example shows how to configuire an interface MAC addres:

```
switch# configure terminal
switch(config)# interface ethernet 3/3
switch(config-if)# mac-address aaaa.bbbb.dddd
switch(config-if)# show interface ethernet 3/3
```

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

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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**mac access-list** *name* **Example:** switch(config)# mac access-list acl-mac-01 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* **Example:** switch(config-mac-acl)# 100 permit mac 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. The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	(Optional) **no** {*sequence-number* | {**permit** | **deny**} *source destination-protocol*} **Example:** switch(config-mac-acl)# no 80	Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic.
Step 5	(Optional) [**no**] **statistics per-entry** **Example:** switch(config-mac-acl)# statistics per-entry	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.

	Command or Action	Purpose

Step 6	(Optional) **show mac access-lists** *name*
	
	Example:
	`switch(config-mac-acl)# show mac`
	`access-lists acl-mac-01`
		Displays the MAC ACL configuration.

Step 7	(Optional) **copy running-config**
	startup-config
	
	Example:
	`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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence mac access-list** *name*
	starting-sequence-number increment
	
	Example:
	`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*
	
	Example:
	`switch(config)# show mac access-lists`
	`acl-mac-01`
		Displays the MAC ACL configuration.

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

Removing a MAC ACL

You can remove a MAC ACL from the device.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** ```switch# configure terminal switch(config)#```	Enters global configuration mode.
Step 2	**no mac access-list** *name* **Example:** ```switch(config)# no mac access-list acl-mac-01 switch(config)#```	Removes the MAC ACL that you specify by name from the running configuration.
Step 3	(Optional) **show mac access-lists** *name* **summary** **Example:** ```switch(config)# show mac access-lists 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** **Example:** ```switch(config)# copy running-config 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** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	Enter one of the following commands:
	    • **interface ethernet** *slot*/*port*
	    • **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		• Enters interface configuration mode for a Layer 2 or Layer 3 interface.
		• Enters interface configuration mode for a Layer 2 or Layer 3 port-channel interface.

Step 3	**mac port access-group** *access-list* **Example:** `switch(config-if)# mac port access-group acl-01`	Applies a MAC ACL to the interface.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	
	Example:
	
	`switch(config)# interface port-channel`
	`5`
	`switch(config-if)#`
		• Enters interface configuration mode for an Ethernet interface.
		
		• Enters interface configuration mode for a port-channel interface.

Step 3	[**no**] **mac packet-classify** **Example:** `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:
	
	• **show running-config interface ethernet** *slot*/*port*
	• **show running-config interface port-channel** *channel-number*
	
	Example:
	
	`switch(config-if)# show running-config`
	`interface ethernet 2/1`
	
	Example:
	
	`switch(config-if)# show running-config`
	`interface port-channel 5`
		• Displays the running configuration of the Ethernet interface.
		
		• Displays the running configuration of the port-channel interface.

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

Verifying the 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. **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. **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 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 *CN93240YC-FX2 NX-OS System Management Configuration Guide*.

Licensing Requirements for VACLs

This table shows the licensing requirements for this feature.

Product	License Requirement
NX-OS	VACLs require no license. Any feature not included in a license package is bundled with the image and is provided at no extra charge to you. For an explanation of the NX-OS licensing scheme, see the *NX-OS Licensing 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 CN93240YC-FX2 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.

- 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.=

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 34: 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vlan access-map** *map-name* [*sequence-number*]
	
	Example:
	
	`switch(config)# vlan access-map`
	`acl-mac-map`
	`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.
		
		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:
	
	    • **match** {**ip** | **ipv6**} **address** *ip-access-list*
	    • **match mac address** *mac-access-list*
	
	Example:
	
	`switch(config-access-map)# match mac`
	`address acl-ip-lab`
	
	Example:
	
	`switch(config-access-map)# match mac`
	`address acl-mac-01`
		Specifies an ACL for the access-map entry.

Step 4	**action** {**drop** | **forward** | **redirect**} **Example:** `switch(config-access-map)# action forward` **Example:**	Specifies the action that the device applies to traffic that matches the ACL. The **action** command supports the **drop**, **forward**, and **redirect** options.

	Command or Action	Purpose
	`switch(config-access-map)# vlan access-map vacl1` `switch(config-access-map)# action redirect e1/1` `switch(config-access-map)# action redirect po100`	
Step 5	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-access-map)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the VACL. The **no** option stops the device from maintaining global statistics for the VACL.
Step 6	(Optional) **show running-config aclmgr** **Example:** `switch(config-access-map)# show running-config aclmgr`	Displays the ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `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** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**no vlan access-map** *map-name* [*sequence-number*] **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** **Example:** switch(config)# show running-config aclmgr	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **vlan filter** *map-name* **vlan-list** *list* **Example:** switch(config)# vlan filter acl-mac-map vlan-list 1-20,26-30 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** **Example:** switch(config)# show running-config aclmgr	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Verifying the 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. **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. **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	*CN93240YC-FX2 NX-OS Quality of Service Configuration Guide*

Configuring Port Security

This chapter describes how to configure port security on 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

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 You cannot configure port security on VXLAN interfaces.

Port Security and Port-Channel Interfaces

Port security is supported on Layer 2 port-channel interfaces. Port security operates on port-channel interfaces in the same manner as on physical interfaces, except as described in this section.

General Guidelines

Port security on a port-channel interface operates in either access mode or trunk mode. In trunk mode, the MAC address restrictions enforced by port security apply to all member ports on a per-VLAN basis.

Enabling port security on a port-channel interface does not affect port-channel load balancing.

Port security does not apply to port-channel control traffic passing through the port-channel interface. Port security allows port-channel control packets to pass without causing security violations. Port-channel control traffic includes the following protocols:

- Port Aggregation Protocol (PAgP)

- Link Aggregation Control Protocol (LACP)

- Inter-Switch Link (ISL)

- IEEE 802.1Q

Configuring Secure Member Ports

The port security configuration of a port-channel interface has no effect on the port security configuration of member ports.

Adding a Member Port

If you add a secure interface as a member port of a port-channel interface, the device discards all dynamic secure addresses learned on the member port but retains all other port-security configuration of the member port in the running configuration. Static secure MAC addresses learned on the secure member port are also stored in the running configuration rather than NVRAM.

If port security is enabled on the member port and not enabled on the port-channel interface, the device warns you when you attempt to add the member port to the port-channel interface. You can use the **force** keyword with the **channel-group** command to forcibly add a secure member port to a nonsecure port-channel interface.

While a port is a member of a port-channel interface, you cannot configure port security on the member port. To do so, you must first remove the member port from the port-channel interface.

Removing a Member Port

If you remove a member port from a port-channel interface, the device restores the port security configuration of the member port. Static secure MAC addresses that were learned on the port before you added it to the port-channel interface are restored to NVRAM and removed from the running configuration.

Note To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

Removing a Port-Channel Interface

If you remove a secure port-channel interface, the following occurs:

- The device discards all secure MAC addresses learned for the port-channel interface, including static secure MAC addresses learned on the port-channel interface.

- The device restores the port-security configuration of each member port. The static secure MAC addresses that were learned on member ports before you added them to the port-channel interface are restored to NVRAM and removed from the running configuration. If a member port did not have port security enabled prior to joining the port-channel interface, port security is not enabled on the member port after the port-channel interface is removed.

| Note | To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports. |

Disabling Port Security

If port security is enabled on any member port, the device does not allow you to disable port security on the port-channel interface. To do so, remove all secure member ports from the port-channel interface first. After disabling port security on a member port, you can add it to the port-channel interface again, as needed.

Port Type Changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

Access Port to Trunk Port

When you change a Layer 2 interface from an access port to a trunk port, the device drops all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN.

Switched Port to Routed Port

When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

Routed Port to Switched Port

When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

Licensing Requirements for Port Security

The following table shows the licensing requirements for this feature:

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

Prerequisites for Port Security

Port security has the following prerequisites:

• You must globally enable port security for the device that you want to protect with port security.

Default Settings for Port Security

This table lists the default settings for port security parameters.

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown

Guidelines and Limitations for Port Security

When configuring port security, follow these guidelines:

- Port security does not support switched port analyzer (SPAN) destination ports.

- Port security does not depend upon other features.

- Port security is not supported on VXLAN interfaces.

- There is no supported method of disabling the USB port on CN93240YC-FX2 switches.

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

Step 2	[**no**] **feature port-security** **Example:** `switch(config)# feature port-security`	Enables port security globally. The **no** option disables port security globally.
Step 3	(Optional) **show port-security** **Example:** `switch(config)# show port-security`	Displays the status of port security.

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

Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

When you disable port security on an interface, all switchport port security configuration for the interface is lost.

Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

Procedure

	Command or Action	**Purpose**

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

	Command or Action	**Purpose**

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the Ethernet or port-channel interface that you want to configure with port security.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

Step 4	[no] **switchport port-security**
	
	Example:
	`switch(config-if)# switchport`
	`port-security`
		Enables port security on the interface. The **no** option disables port security on the interface.

Step 5	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Enabling or Disabling Sticky MAC Address Learning

You can disable or enable sticky MAC address learning on an interface. If you disable sticky learning, the device returns to dynamic MAC address learning on the interface, which is the default learning method.

By default, sticky MAC address learning is disabled.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	**Purpose**

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

	Command or Action	Purpose

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with sticky MAC address learning.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.

Step 4	[**no**] **switchport port-security mac-address sticky**
	
	Example:
	`switch(config-if)# switchport`
	`port-security mac-address sticky`
		Enables sticky MAC address learning on the interface. The **no** option disables sticky MAC address learning.

Step 5	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Adding a Static Secure MAC Address on an Interface

You can add a static secure MAC address on a Layer 2 interface.

Note If the MAC address is a secure MAC address on any interface, you cannot add it as a static secure MAC address to another interface until you remove it from the interface on which it is already a secure MAC address.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

You must have enabled port security globally.

Verify that the interface maximum has not been reached for secure MAC addresses. If needed, you can remove a secure MAC address, or you can change the maximum number of addresses on the interface.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	　• **interface ethernet** *slot*/*port*
	　• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you specify.

Step 3	[**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
	
	Example:
	`switch(config-if)# switchport`
	`port-security mac-address 0019.D2D0.00AE`
		Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Removing a Static Secure MAC Address on an Interface

You can remove a static secure MAC address on a Layer 2 interface.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands: 　• **interface ethernet** *slot*/*port*	Enters interface configuration mode for the interface from which you want to remove a static secure MAC address.

	Command or Action	Purpose

	• **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`	

Step 3	**no switchport port-security mac-address** *address*
	
	Example:
	
	`switch(config-if)# no switchport`
	`port-security mac-address 0019.D2D0.00AE`
		Removes the static secure MAC address from port security on the current interface.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Removing a Sticky Secure MAC Address

You can remove a sticky secure MAC address, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	
	  • **interface ethernet** *slot*/*port*
	  • **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a sticky secure MAC address.

	Command or Action	Purpose
Step 3	**no switchport port-security mac-address sticky** **Example:** `switch(config-if)# no switchport port-security mac-address sticky`	Disables sticky MAC address learning on the interface, which converts any sticky secure MAC addresses on the interface to dynamic secure MAC addresses.
Step 4	**clear port-security dynamic address** *address* **Example:** `switch(config-if)# clear port-security dynamic address 0019.D2D0.02GD`	Removes the dynamic secure MAC address that you specify.
Step 5	(Optional) **show port-security address interface** {**ethernet** *slot*/*port*	**port-channel** *channel-number*} **Example:** `switch(config)# show port-security address interface ethernet 2/1`	Displays secure MAC addresses. The address that you removed should not appear.
Step 6	(Optional) **switchport port-security mac-address sticky** **Example:** `switch(config-if)# switchport port-security mac-address sticky`	Enables sticky MAC address learning again on the interface.

Removing a Dynamic Secure MAC Address

You can remove dynamically learned, secure MAC addresses.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	**clear port-security dynamic** {**interface ethernet** *slot*/*port*	**address** *address*} [**vlan** *vlan-ID*] **Example:** `switch(config)# clear port-security dynamic interface ethernet 2/1`	Removes dynamically learned, secure MAC addresses, as specified. If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify.

	Command or Action	**Purpose**
		If you use the **address** keyword, you remove the single, dynamically learned address that you specify.
		Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.
Step 3	(Optional) **show port-security address** **Example:** `switch(config)# show port-security address`	Displays secure MAC addresses.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of addresses is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Note When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the device rejects the command. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	**Purpose**

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

	Command or Action	Purpose

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot* is the interface that you want to configure with the maximum number of MAC addresses.

Step 3	[**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]
	
	Example:
	`switch(config-if)# switchport`
	`port-security maximum 425`
		Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 1025. The **no** option resets the maximum number of MAC addresses to the default, which is 1.
		
		If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Configuring an Address Aging Type and Time

You can configure the MAC address aging type and the length of time that the device uses to determine when MAC addresses learned by the dynamic method have reached their age limit.

Absolute aging is the default aging type.

By default, the aging time is 0 minutes, which disables aging.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	Enter one of the following commands:
	
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with the MAC aging type and time.

Step 3	[**no**] **switchport port-security aging type** {**absolute** | **inactivity**}
	
	Example:
	`switch(config-if)# switchport`
	`port-security aging type inactivity`
		Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging.

Step 4	[**no**] **switchport port-security aging time** *minutes*
	
	Example:
	`switch(config-if)# switchport`
	`port-security aging time 120`
		Configures the number of minutes that a dynamically learned MAC address must age before the device drops the address. The maximum valid *minutes* is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).

Step 5	(Optional) **show running-config port-security**
	
	Example:
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Configuring a Security Violation Action

You can configure the action that the device takes if a security violation occurs. The violation action is configurable on each interface that you enable with port security.

The default security action is to shut down the port on which the security violation occurs.

Before you begin

You must have enabled port security globally.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	  • **interface ethernet** *slot*/*port*
	  • **interface port-channel** *channel-number*
	
	Example:
	
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with a security violation action.

Step 3	[**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**}
	
	Example:
	
	`switch(config-if)# switchport`
	`port-security violation restrict`
		Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface.

Step 4	(Optional) **show running-config port-security**
	
	Example:
	
	`switch(config-if)# show running-config`
	`port-security`
		Displays the port security configuration.

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

Verifying the Port Security Configuration

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

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status of the device.
show port-security interface	Displays the port security status of a specific interface.

Command	Purpose
show port-security address	Displays secure MAC addresses.
show vpc consistency-parameters vpc id	Verifies configuration on both vPC peers.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses.

Configuration Example for Port Security

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

Configuration Examples for Port Security in a vPC Domain

The following example shows how to enable and configure port security on vPC peers in a vPC domain. The first switch is the primary vPC peer and the second switch is the secondary vPC peer. Before configuring port security on the switches, create the vPC domain and check that the vPC peer-link adjacency is established.

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	*CN93240YC-FX2 NX-OS Layer 2 Switching Configuration Guide*

MIBs

NX-OS provides read-only SNMP support for port security.

Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a 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 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 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 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 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 NX-OS device. The NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The 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 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
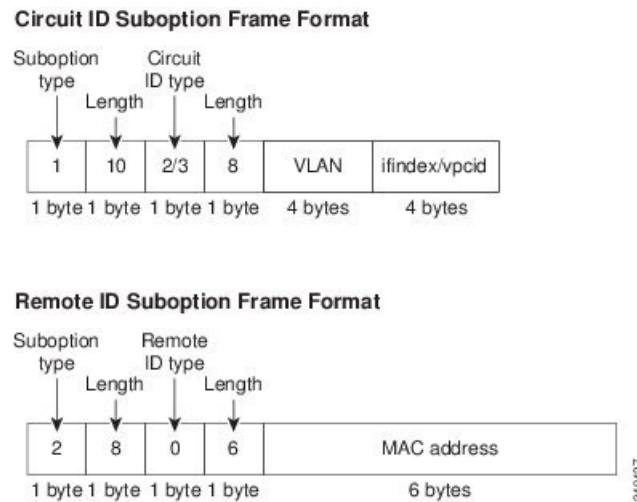
About the DHCP Relay Agent

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 10: 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.
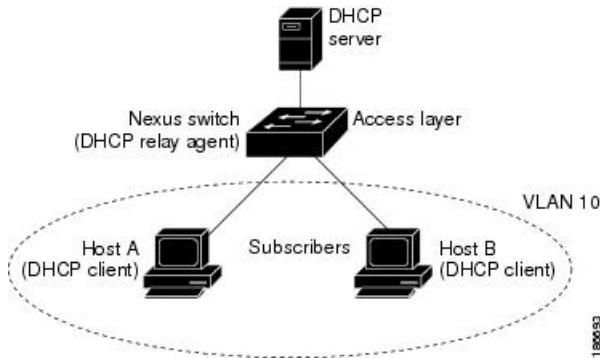
When you enable Option 82 for the DHCP relay agent on the NX-OS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.

2. When the 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 NX-OS device if the request was relayed to the server by the device. The NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The 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 11: Suboption Packet Formats

Circuit ID Suboption Frame Format (for non-VXLAN VLANs)
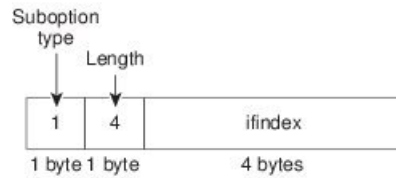
Circuit ID Suboption Frame Format (for VXLAN VLANs)
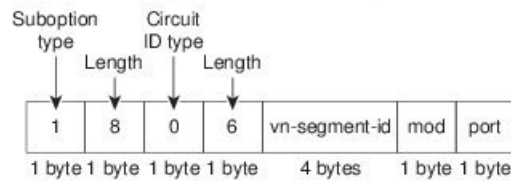
Remote ID Suboption Frame Format
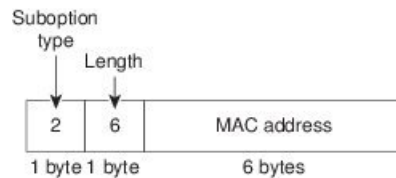
VRF Support for the DHCP Relay Agent

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 *CN93240YC-FX2 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 *CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*.

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).

Licensing Requirements for DHCP

This table shows the licensing requirements for DHCP.

Product	License Requirement
NX-OS	DHCP 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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.

• CN93240YC-FX2 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 Nexus device).

• 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.

- In a vPC environment with DHCP smart relay enabled, the subnet of the primary and secondary addresses of an interface should be the same on both Cisco NX-OS devices.

- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.

- 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.=

Note   For DHCP configuration limits, see the *CN93240YC-FX2 NX-OS Verified Scalability Guide*.

Default Settings for DHCP

This table lists the default settings for DHCP parameters.

Table 35: Default DHCP Parameters

Parameters	Default
DHCP feature	Disabled
DHCP snooping	Disabled

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

Step 2	[**no**] **feature dhcp** **Example:** `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** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

	Command or Action	Purpose
Step 2	[**no**] **ip dhcp snooping** **Example:** `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** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list*
	
	Example:
	
	`switch(config)# ip dhcp snooping vlan`
	`100,200,250-252`
		Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** form of this command disables DHCP snooping on the VLANs specified.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

	Command or Action	Purpose

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

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**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping verify mac-address**
	
	Example:
	`switch(config)# ip dhcp snooping verify`
	`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** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

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

Enabling or Disabling 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping information option**
	
	Example:
	
	`switch(config)# ip dhcp snooping`
	`information option`
		Enables the insertion and removal of Option 82 information for DHCP packets. The **no** 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*
	
	Example:
	
	`switch(config)# ip dhcp snooping`
	`sub-option circuit-id format-type string`
	` format`
	
	Example:
	
	`switch(config)# ip dhcp snooping`
	`sub-option circuit-id format-type string`
	` format?`
	`WORD Format string (Max Size 64)`
		Configures Option 82 as follows:
		
		• If you do not specify *format-type*, the *circuit-id* displays the incoming port, for example, *ethernet1/1*.
		
		• If you specify format *<word>*, the *circuit-id* displays the specified word
		
		• If you specify *%h* instead of *<word>*, the *circuit-id* displays the host name.
		
		• If you specify *%p* instead of *<word>*, the *circuit-id* displays the port name.

	Command or Action	Purpose
		• If you specify *%h:%p* instead of *<word>*, the *circuit-id* displays both host and port name.
		Note The *no* option disables this behavior.

Step 4	**interface** *interface slot/port*
	
	 Example:
	 `switch(config)# interface ethernet 2/2`
	 `switch(config-if)#`
		Enters the interface configuration mode, where slot/port is the interface where you want to enable or disable snooping.

Step 5	(Optional) **ip dhcp option82 sub-option circuit-id**
	
	 Example:
	 `switch(config-if)# ip dhcp option82`
	 `sub-option circuit-id?`
	 `WORD Format string (Max Size 64)`
	
	 Example:
	 `switch(config-if)# ip dhcp option82`
	 `sub-option circuit-id test`
	 `switch(config-if)#`
		Configures Option 82 at the interface.
		
		 Note The *no* option disables this behavior

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

Step 7	(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 8	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

Step 9	(Optional) **copy running-config startup-config**
	
	 Example:
	 `switch(config)# copy running-config`
	 `startup-config`
		Copies the running configuration to the startup configuration.

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

Step 2	[**no**] **ip dhcp packet strict-validation**
	
	Example:
	`switch(config)# ip dhcp packet`
	`strict-validation`
		Enables the strict validation of DHCP packets. The **no** form of this command disables strict DHCP packet validation.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

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

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

Step 2	Do one of the following options: - **interface ethernet** *slot*/*port* - **interface port-channel** *channel-number*	- 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.

	Command or Action	Purpose

	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		• 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** **Example:** `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** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling 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**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay information option trust** **Example:** `switch(config)# ip dhcp relay information 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** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.

	Command or Action	**Purpose**
Step 4	(Optional) **show ip dhcp relay information trusted-sources** **Example:** `switch(config)# show ip dhcp relay information trusted-sources`	Displays the DHCP relay trusted ports configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 2	**interface** [**ethernet** *slot/port*[*.number*] | **port-channel** *channel-number*]
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`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** **Example:** `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.

	Command or Action	Purpose
		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.
Step 4	(Optional) **show ip dhcp relay information trusted-sources** **Example:** `switch(config-if)# show ip dhcp relay information trusted-sources`	Displays the DHCP relay trusted ports configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 2	[**no**] **ip dhcp relay information trust-all**
	
	Example:
	
	`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.

Step 3	(Optional) **show ip dhcp relay information trusted-sources**
	
	Example:
	
	`switch(config)# show ip dhcp relay`
	`information trusted-sources`
		Displays the DHCP relay trusted ports configuration.

Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

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

Enabling or Disabling 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay** **Example:**	Enables the DHCP relay agent. The **no** option disables the DHCP relay agent.

	Command or Action	Purpose
	switch(config)# ip dhcp relay	
Step 3	(Optional) **show ip dhcp relay** **Example:** switch(config)# show ip dhcp relay	Displays the DHCP relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

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) 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. For VLANs and SVIs:

	Command or Action	Purpose
		• 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.
		• 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 5	(Optional) switch(config)# **show ip dhcp relay**	Displays the DHCP relay configuration.
Step 6	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP configuration.
Step 7	(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** **Example:** ```switch# configure terminal switch(config)#```	Enters global configuration mode.
Step 2	[**no**] **ip dhcp relay information option vpn** **Example:**	Enables VRF support for the DHCP relay agent. The **no** option disables this behavior.

	Command or Action	Purpose
	switch(config)# ip dhcp relay information option vpn	
Step 3	[no] ip dhcp relay sub-option type cisco **Example:** 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** **Example:** switch(config)# show ip dhcp relay	Displays the DHCP relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Disabling the Server Identifier Override Option

Beginning with 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** **Example:** switch# **configure terminal**	Enters global configuration mode.
Step 2	[no] **ip dhcp relay information option server-id-override-disable** **Example:**	Disables the server identifier override option in DHCP Option 82 packets.

Command or Action	Purpose	
switch(config)# **ip dhcp relay information option server-id-override-disable**	**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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	
	• **interface ethernet** *slot*/*port*[.*number*]
	• **interface vlan** *vlan-id*
	• **interface port-channel** *channel-id*[.*subchannel-id*]
	
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.
		
		Note Port-channel subinterfaces are supported only in NX-OS Releases 6.1(2)I3(3) and 6.1(2)I3(3a). They are not supported in NX-OS Release 9.2(1).

	Command or Action	Purpose
		• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCP server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.
Step 3	**ip dhcp relay address** *IP-address* [**use-vrf** *vrf-name*] **Example:** `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. To configure more than one IP address, use the **ip dhcp relay address** command once per address.
Step 4	(Optional) **show ip dhcp relay address** **Example:** `switch(config-if)# show ip dhcp relay address`	Displays all the configured DHCP server addresses.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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.

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

Step 2	[**no**] **ip dhcp relay source-interface** *interface*
	
	Example:
	
	`switch(config)# ip dhcp relay`
	`source-interface loopback 2`
		Configures the source interface for the DHCP relay agent.
		
		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*] **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.

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

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** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	[no] ip dhcp smart-relay global **Example:** 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** **Example:** switch(config)# show ip dhcp relay	Displays the DHCP smart relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Enabling or Disabling 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** **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 DHCP smart relay.

	Command or Action	**Purpose**
Step 3	[**no**] **ip dhcp smart-relay** **Example:** `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**
	
	Example:
	
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

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

Step 6	(Optional) **show ip dhcp relay** **Example:** `switch# show ip dhcp relay`	Displays the DHCP smart relay configuration.
Step 7	(Optional) **show running-config dhcp** **Example:** `switch# show running-config dhcp`	Displays the DHCP configuration.

Step 8	(Optional) **copy running-config startup-config**
	
	Example:
	
	`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 used in subnet-selection for source and relay agent. In the case of a VPN or a source interface option, the option82 link selection is updated with the configured subnet.

The DHCP relay and the subnet-selection configuration are mutually exclusive at the interface level. If DHCP 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 option82 link-selection to assign the IP address.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**interface vlan** *vlan-id* **Example:** switch(config)#interface vlan 3 switch(config-if)#	Enters interface configuration mode.
Step 3	**ip dhcp relay subnet-selection** *ip address* **Example:** switch(config-if)#ip dhcp relay 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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ipv6 dhcp relay** **Example:** switch(config)# ipv6 dhcp relay	Enables the DHCPv6 relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*]	Displays the DHCPv6 relay configuration.

	Command or Action	Purpose
	Example: switch(config)# show ipv6 dhcp relay	
Step 4	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Enabling or Disabling VRF Support for the DHCPv6 Relay Agent

You can configure the device to support the relaying of DHCPv6 requests that arrive on an interface in one VRF to a DHCPv6 server in a different VRF.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ipv6 dhcp relay option vpn** **Example:** switch(config)# ipv6 dhcp relay option vpn	Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior.
Step 3	[**no**] **ipv6 dhcp relay option type cisco** **Example:** switch(config)# ipv6 dhcp relay option type cisco	Causes the DHCPv6 relay agent to insert virtual subnet selection (VSS) details as part of the vendor-specific option. The **no** option causes the DHCPv6 relay agent to insert VSS details as part of the VSS option (68), which is defined in RFC-6607. This command is useful when you want to use DHCPv6 servers that do not support RFC-6607 but allocate IPv6 addresses based on the client VRF name.

	Command or Action	Purpose
Step 4	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

Step 2	Do one of the following options:
	    • **interface ethernet** *slot/port*
	    • **interface port-channel** *channel-id*
		• Enters interface configuration mode, where *slot/port* is the physical Ethernet interface that you want to configure with a DHCPv6 server IP address.

	Command or Action	Purpose

	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• 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*]
	
	Example:
	`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.
		
		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.
		
		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.
		
		To configure more than one IP address, use the **ipv6 dhcp relay address** command once per address.

Step 4	(Optional) **show running-config dhcp**
	
	Example:
	`switch(config-if)# show running-config`
	`dhcp`
		Displays the DHCPv6 configuration.

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

Enabling DHCPv6 Option 79

Beginning with 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** **Example:** switch# **configure terminal**	Enters global configuration mode.
Step 2	**ipv6 dhcp relay option79** **Example:** 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. 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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ipv6 dhcp relay source-interface** *interface* **Example:** switch(config)# ipv6 dhcp relay source-interface loopback 2	Configures the source interface for the DHCPv6 relay agent. **Note** The DHCPv6 relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*]	Displays the DHCPv6 relay configuration.

	Command or Action	Purpose
	Example: switch(config)# show ipv6 dhcp relay	
Step 4	(Optional) **show running-config dhcp show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

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

Step 2	Do one of the following options:
	
	  • **interface ethernet** *slot*/*port*
	  • **interface mgmt 0**
	  • **interface vlan** *vlan-id*
	
	Example:
	`switch(config)# interface vlan 3`
	`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.
		
		• Enters interface configuration mode and specifies the management interface as the interface for which you want to enable the DHCP client feature.
		
		• 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**
	
	Example:
	`switch(config-if)# ipv6 address`
	`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.

	Command or Action	Purpose
Step 4	[**no**] {**ip** | **ipv6**} **address dhcp** **Example:** `switch(config-if)# ip address dhcp`	Assigns an IPv4 or IPv6 address to the interface. The **no** form of this command releases the IP address.
Step 5	(Optional) Do one of the following options: • **show running-config interface ethernet** *slot*/*port* • **show running-config interface mgmt 0** • **show running-config interface vlan** *vlan-id* **Example:** `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** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration. Only the {**ip** | **ipv6**} **address dhcp** command is saved. The assigned IP address is not saved even though it shows in the running configuration.

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.
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.

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)	*CN93240YC-FX2 NX-OS Security Configuration Guide*
IP Source Guard	*CN93240YC-FX2 NX-OS Security Configuration Guide*
vPCs	*CN93240YC-FX2 NX-OS Interfaces Configuration Guide*
VRFs and Layer 3 virtualization	*CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*

Standards

Standards	Title
RFC 2131	Dynamic Host Configuration Protocol (http://tools.ietf.org/html/rfc2131)
RFC 3046	DHCP Relay Agent Information Option (http://tools.ietf.org/html/rfc3046)
RFC 6607	Virtual Subnet Selection Options for DHCPv4 and DHCPv6 (http://tools.ietf.org/html/rfc6607)
RFC 6939	Client Link-Layer Address Option in DHCPv6 (https://tools.ietf.org/html/rfc6939)

Configuring IPv6 First Hop Security

This chapter describes how to configure First Hop Security (FHS) features on NX-OS devices.

This chapter includes the following sections:

- Introduction to First-Hop Security
- Guidelines and Limitations of First Hop Security
- About vPC First Hop Security Configuration
- RA Guard
- DHCPv6 Guard
- IPv6 Snooping
- How to Configure IPv6 FHS
- Configuration Examples
- Additional References for IPv6 First-Hop Security=

Introduction to First-Hop Security

The Layer 2 and Layer 3 switches operate in the Layer 2 domains with technologies such as server virtualization, Overlay Transport Virtualization (OTV), and Layer 2 mobility. These devices are sometimes referred to as "first hops", specifically when they are facing end nodes. The First-Hop Security feature provides end node protection and optimizes link operations on IPv6 or dual-stack networks.

First-Hop Security (FHS) is a set of features to optimize IPv6 link operation, 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 408 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.

All port level FHS policies are programmed in the ifacl region, while the VLAN level policies are programmed in the FHS region. Use the hardware profile **tcam regionfhs tcam_size** command to configure the FHS. The range for the TCAM size is 0-4096.

- CN93240YC-FX2 platform switches, FHS packets take the copp-s-dhcpreq queue for software processing.=

Note   When you upgrade the Cisco Nexus Series switch to NX-OS Release 7.0(3)I7(1) using the In-Service Software Upgrades (ISSU), you must reload the 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, we recommend carving the **ing-redirect** TCAM region on CN93240YC-FX2 platform 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 12: FHS Configuration with DHCP relay on-stack
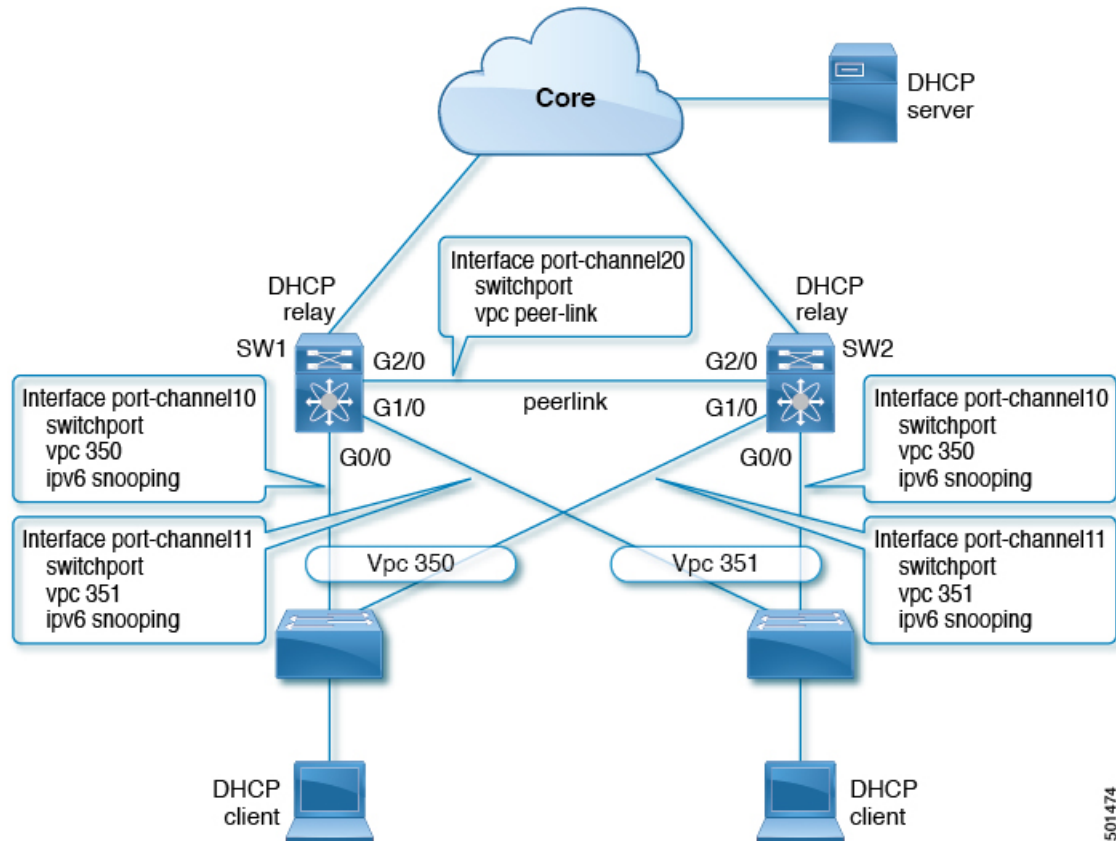
In the figure, snooping policy is enabled on both vPC links. In this scenario, the two vPC peers learn all the host IP/MAC bindings behind the vPC links and sync these up between themselves. The two vPC peers learn the bindings using both IPv6 ND and IPv6 DHCP control protocols.

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 13: FHS Configuration with external DHCP relay
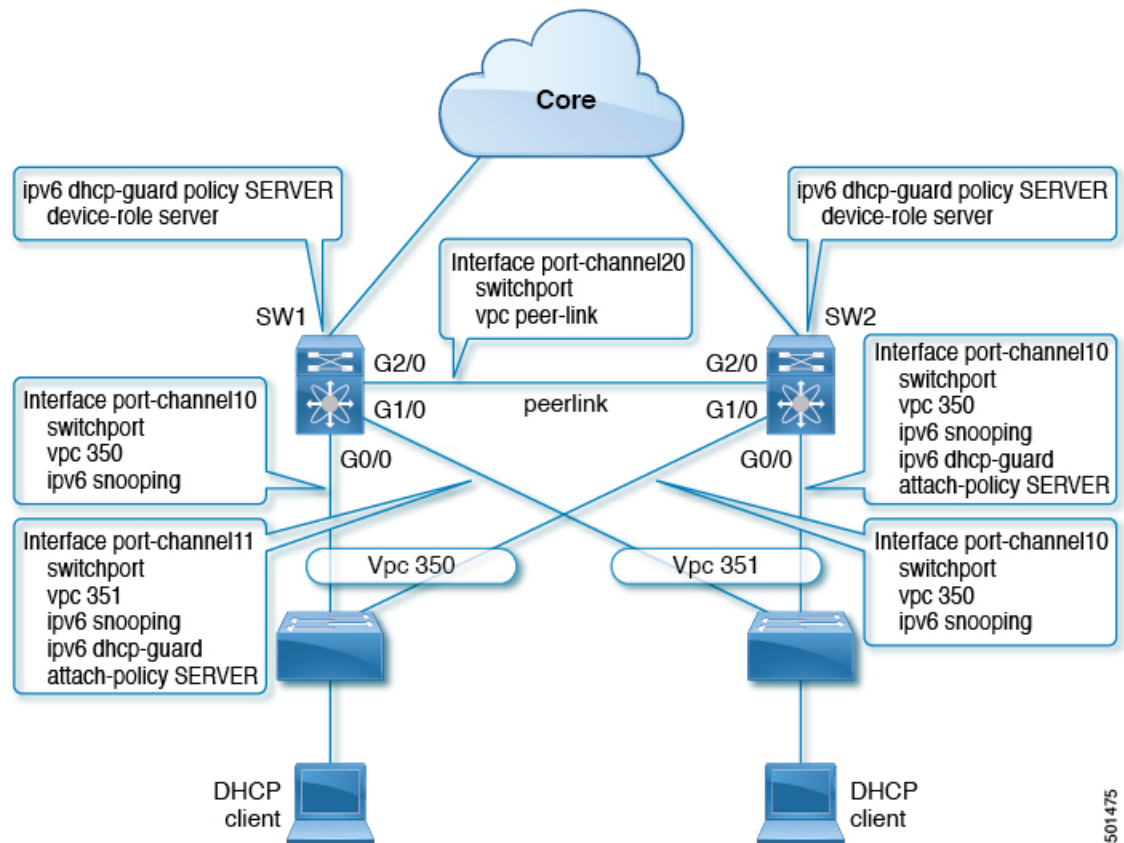
In the figure, the clients are located behind the vPC links with the default IPv6 snooping policy. You can attach both ipv6 snooping and ipv6 dhcp-guard attach-policy SERVER policies to the links where DHCP server traffic arrives. You will need both the server or relay facing and client facing IPv6 snooping 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 14: FHS configuration with client and DHCP relay on orphan port
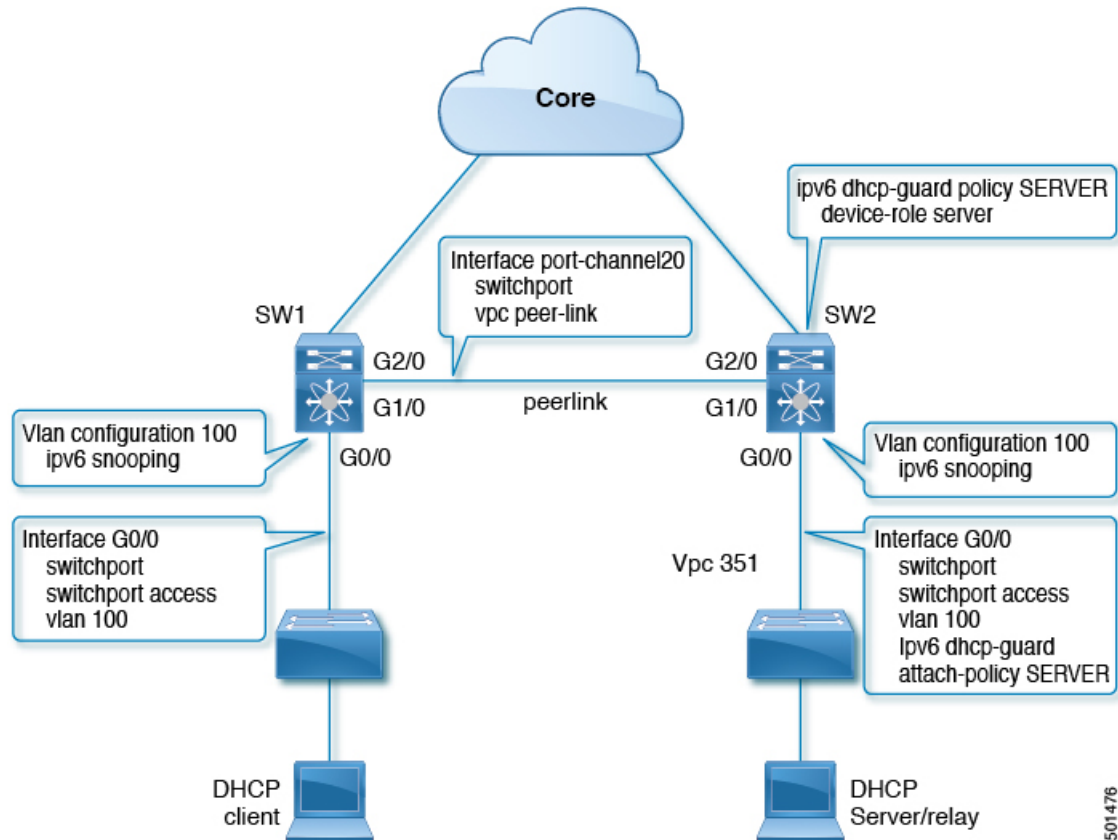
RA Guard

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.

Guidelines and Limitations of IPv6 RA Guard

The guidelines and limitations of IPv6 RA Guard are as follows:

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.

- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.

- This feature can be configured on a switch port interface in the ingress direction.

- This feature supports host mode and router mode.

- This feature is supported only in the ingress direction; it is not supported in the egress direction.

- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.

- Packets dropped by the IPv6 RA Guard feature can be spanned.

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** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy policy1`	Defines the RA guard policy name and enters RA guard policy configuration mode.
Step 3	**device-role** {**host**	**router**	**monitor**	**switch**} **Example:** `Device(config-ra-guard)# device-role router`	Specifies the role of the device attached to the port. • 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. • 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. • 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.

	Command or Action	Purpose
		• 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*} **Example:** Device(config-ra-guard)# hop-limit minimum 3	(Optional) Enables verification of the advertised hop count limit. • If not configured, this check will be bypassed.
Step 5	**managed-config-flag** {**on**	**off**} **Example:** Device(config-ra-guard)# managed-config-flag on	(Optional) Enables verification that the advertised managed address configuration flag is on. • If not configured, this check will be bypassed.
Step 6	**other-config-flag** {**on**	**off**} **Example:** Device(config-ra-guard)# other-config-flag on	(Optional) Enables verification of the advertised "other" configuration parameter.
Step 7	**router-preference maximum** {**high**	**low**	**medium**} **Example:** Device(config-ra-guard)# router-preference maximum high	(Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit.
Step 8	**trusted-port** **Example:** Device(config-ra-guard)# trusted-port	(Optional) Specifies that this policy is being applied to trusted ports. • All RA guard policing will be disabled.
Step 9	**exit** **Example:** Device(config-ra-guard)# exit	Exits RA guard policy configuration mode and returns to global configuration mode.

Configuring IPv6 RA Guard on an Interface

Procedure

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

	Command or Action	Purpose

Step 2	**interface** *type* *number*
	Example:
	`Device(config)# interface ethernet 1/1`
	Example:
	`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*] **Example:** `Device(config-if)# ipv6 nd raguard attach-policy`	Applies the IPv6 RA Guard feature to a specified interface.
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Step 5	**show ipv6 nd raguard policy** [*policy-name*]
	Example:
	`switch# show ipv6 nd raguard policy host`
	`Policy host configuration:`
	`  device-role host`
	
	`Policy applied on the following`
	`interfaces:`
	
	`  Et0/0        vlan all`
	`  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*] **Example:** `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** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	**ipv6 dhcp guard policy** *policy-name*	Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode.
	Example:	
	Device(config)# ipv6 dhcp guard policy pol1	
Step 3	**device-role** {**client** | **server**}	Specifies the device role of the device attached to the target (interface or VLAN).
	Example:	• device-role client—Interface where a normal DHCPv6 client is connected. It blocks any incoming server packets.
	Device(config-dhcp-guard)# device-role server	• device-role server—Interface where a normal DHCPv6 server is connected. It allows all DHCPv6 packets originating on this interface.
Step 4	**preference min** *limit*	(Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed.
	Example:	
	Device(config-dhcp-guard)# preference min 0	
Step 5	**preference max** *limit*	(Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed.
	Example:	
	Device(config-dhcp-guard)# preference max 255	
Step 6	**trusted-port**	(Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled.
	Example:	
	Device(config-dhcp-guard)# trusted-port	
Step 7	**exit**	Exits DHCP guard configuration mode and returns to global configuration mode.
	Example:	
	Device(config-dhcp-guard)# exit	
Step 8	**interface** *type number*	Specifies an interface and enters interface configuration mode.
	Example:	
	Device(config)# interface GigabitEthernet 0/2/0	
Step 9	**switchport**	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
	Example:	

	Command or Action	Purpose
	`Device(config-if)# switchport`	
Step 10	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** `Device(config-if)# ipv6 dhcp guard attach-policy pol1`	Attaches a DHCPv6 guard policy to an interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**vlan configuration** *vlan-id* **Example:** `Device(config)# vlan configuration 1`	Specifies a VLAN and enters VLAN configuration mode.
Step 13	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** `Device(config-vlan-config)# ipv6 dhcp guard attach-policy pol1`	Attaches a DHCPv6 guard policy to a VLAN.
Step 14	**exit** **Example:** `Device(config-vlan-config)# exit`	Exits VLAN configuration mode and returns to global configuration mode.
Step 15	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 16	**show ipv6 dhcp guard policy** [*policy-name*] **Example:** `Device# show ipv6 dhcp policy guard pol1`	(Optional) Displays the policy configuration as well as all the interfaces where the policy is applied.

Configuring IPv6 Snooping

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 snooping policy** *policy-name* **Example:** `Device(config)# ipv6 snooping policy policy1`	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.
Step 3	**device-role** { **node** | **switch** } **Example:** `Device(config-snoop-policy)# device-node switch`	Specifies the role of the device attached to the target (interface or VLAN): • node—is the default. Bindings are created and entries are probed. • switch—Entries are not probed and when a trusted port is enabled, bindings are not created.
Step 4	**[no] limit address-count** **Example:** `Device(config-snoop-policy)# limit address-count 500`	Limits the number of binding entries, a no limit address-count means no limit.

Step 5	**[no] protocol** *dhcp* | *ndp*
	
	Example:
	
	`Device(config-snoop-policy)# protocol dhcp`
	
	`Device(config-snoop-policy)# protocol ndp`
		Turns on or switches off either DHCP or NDP gleaning.

Step 6	**trusted-port** **Example:** `Device(config-snoop-policy)# trusted-port`	Specifies that the policy be applied to a trusted port. If an entry is a trusted-port, none of it's traffic will be blocked or dropped.
Step 7	**security-level** *glean* | *guard* | *inspect* **Example:** `Device(config-snoop-policy)# security-level guard`	Specifies the type of security applied to the policy: glean, guard, or inspect. Here is what each security level means: • glean—learns bindings but does not drop packets.

	Command or Action	Purpose
		• inspect—learns bindings and drops packets in case it detects an issue, such as address theft. • guard—works like inspect, but in addition drops IPv6, ND, RA, and IPv6 DHCP Server packets in case of a threat.
Step 8	**tracking** **Example:** Device(config-snoop-policy)# tracking enable	Enables tracking.
Step 9	**exit** **Example:** Device(config-snoop-policy)# exit	Exits snooping configuration mode and returns to global configuration mode.
Step 10	**interface** *type-number* **Example:** Device(config-if)# interface ethernet 1/25	Specifies an interface and enters interface configuration mode.
Step 11	**[no] switchport** **Example:** Device(config-if)# switchport	Switches between Layer 2 and Layer 3 mode.
Step 12	**ipv6 snooping attach-policy** *policy-name* **Example:** Device(config-if)# ipv6 snooping attach-policy policy1	Attaches the IPv6 snooping policy to an interface.
Step 13	**exit** **Example:** Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 14	**vlan configuration** *vlan-id* **Example:** Device(config)# vlan configuration 333	Specifies a VLAN and enters VLAN configuration mode.
Step 15	**ipv6 snooping attach-policy** *policy-name* **Example:** Device(config-vlan-config)# ipv6 snooping attach-policy policy1	Attaches the IPv6 snooping policy to a VLAN.
Step 16	**exit** **Example:** Device(config-vlan-config)# exit	Exits VLAN configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 17	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 18	**show ipv6 snooping policy** *policy-name* **Example:** `Device(config)# show ipv6 snooping policy policy1`	Displays the policy configuration and the interfaces where the policy is applied.

Verifying and Troubleshooting IPv6 Snooping

Procedure

	Command or Action	Purpose
Step 1	**show ipv6 snooping capture-policy** [**interface** *type number*] **Example:** `Device# show ipv6 snooping capture-policy interface ethernet 0/0`	Displays snooping message capture policies.
Step 2	**show ipv6 snooping counter** [**interface** *type number*] **Example:** `Device# show ipv6 snooping counter interface FastEthernet 4/12`	Displays information about the packets counted by the interface counter.
Step 3	**show ipv6 snooping features** **Example:** `Device# show ipv6 snooping features`	Displays information about snooping features configured on the device.
Step 4	**show ipv6 snooping policies** [**interface** *type number*] **Example:** `Device# show ipv6 snooping policies`	Displays information about the configured policies and the interfaces to which they are attached.
Step 5	**debug ipv6 snooping** **Example:** `Device# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

Configuration Examples

Example: IPv6 RA Guard Configuration

```
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
NX-OS Licensing	*NX-OS Licensing Guide*

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on a NX-OS device.

This chapter includes the following sections:

- About DAI
- Licensing Requirements for DAI
- Prerequisites for DAI
- Guidelines and Limitations for DAI
- Default Settings for DAI
- Configuring DAI
- Verifying the DAI Configuration
- Monitoring and Clearing DAI Statistics
- Configuration Examples for DAI
- Additional References for DAI=

About DAI

ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

ARP Spoofing Attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your Layer 2 network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic intended for other hosts on the subnet.

Figure 15: ARP Cache Poisoning

This figure shows an example of ARP cache poisoning.
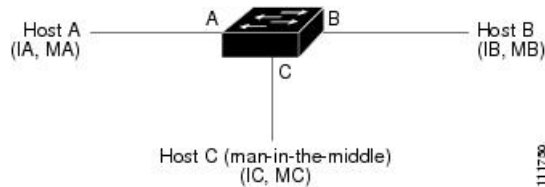
Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address associated with IP address IB. When the device and host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds, the device and host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the device, host A, and host B by broadcasting two forged ARP responses with bindings: one for a host with an IP address of IA and a MAC address of MC and another for a host with an IP address of IB and a MAC address of MC. Host B and the device then use the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Likewise, host A and the device use the MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a *man-in-the middle* attack.

DAI and ARP Spoofing Attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco Nexus device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports

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

- Drops invalid ARP packets

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database 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 16: 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.
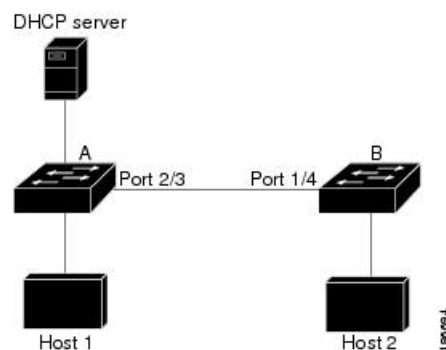
If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though 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

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 CN 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   NX-OS does not generate system messages about DAI packets that are logged.

Licensing Requirements for DAI

This table shows the licensing requirements for DAI.

Product	License Requirement
NX-OS	DAI 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

Prerequisites for DAI

- You must enable the DHCP feature before you can configure DAI. See Configuring DHCP..

- You must configure the VLANs on which you want to enable DAI. See the *CN93240YC-FX2 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.=

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 onfigures 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.=

Default Settings for DAI

This table lists the default settings for DAI parameters.

Table 36: 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** **Example:** ``` switch# configure terminal switch(config)# ```	Enters global configuration mode.

	Command or Action	Purpose
Step 2	[**no**] **ip arp inspection vlan** *vlan-list* **Example:** `switch(config)# ip arp inspection vlan 13`	Enables DAI for the specified list of VLANs. The **no** option disables DAI for the specified VLANs.
Step 3	(Optional) **show ip arp inspection vlan** *vlan-id* **Example:** `switch(config)# show ip arp inspection vlan 13`	Displays the DAI configuration for a specific VLAN.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the DAI Trust State of a Layer 2 Interface

You can configure the DAI interface trust state of a Layer 2 interface. By default, all interfaces are untrusted.

A device forwards ARP packets that it receives on a trusted Layer 2 interface but does not check them.

On untrusted interfaces, the device intercepts all ARP requests and responses and verifies that the intercepted packets have valid IP-MAC address bindings before updating the local cache and forwarding the packet to the appropriate destination. If the device determines that packets have invalid bindings, it drops the packets and logs them according to the logging configuration.

Before you begin

If you are enabling DAI, make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose
Step 3	[**no**] **ip arp inspection trust** **Example:** `switch(config-if)# ip arp inspection trust`	Configures the interface as a trusted ARP interface. The **no** option configures the interface as an untrusted ARP interface.
Step 4	(Optional) **show ip arp inspection interface** *type port/slot* **Example:** `switch(config-if)# show ip arp inspection interface ethernet 2/1`	Displays the trust state and the ARP packet rate for the specified interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Additional Validation

You can enable or disable additional validation of ARP packets. By default, no additional validation of ARP packets is enabled. When no additional validation is configured, the source MAC address and the source IP address check against the IP-to-MAC binding entry for ARP packets is performed by using the Ethernet source MAC address (not the ARP sender MAC address) and the ARP sender IP address.

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can use the following keywords with the **ip arp inspection validate** command to implement additional validations:

dst-mac

> Checks the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

ip

> Checks the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

src-mac

> Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, follow these guidelines:

- You must specify at least one of the keywords. You can specify one, two, or all three keywords.

• Each **ip arp inspection validate** command that you enter replaces the configuration from any previous commands. If you enter an **ip arp inspection validate** command to enable src-mac and dst-mac validations, and a second **ip arp inspection validate** command to enable ip validation, the src-mac and dst-mac validations are disabled when you enter the second command.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}
	
	Example:
	
	`switch(config)# ip arp inspection`
	`validate src-mac dst-mac ip`
		Enables additional DAI validation. The **no** form of this command disables additional DAI validation.

Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.

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

Configuring the DAI Logging Buffer Size

You can configure the DAI logging buffer size. The default buffer size is 32 messages.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip arp inspection log-buffer entries** *number*
	
	Example:
	
	`switch(config)# ip arp inspection`
	`log-buffer entries 64`
		Configures the DAI logging buffer size. The **no** option reverts to the default buffer size, which is 32 messages. The buffer size can be between 1 and 1024 messages.

	Command or Action	Purpose
Step 3	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Configuring DAI Log Filtering

You can configure how the device determines whether to log a DAI packet. By default, the device logs DAI packets that are dropped.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**} **Example:** switch(config)# ip arp inspection vlan 100 dhcp-bindings permit	Configures DAI log filtering, as follows. The **no** form of this command removes DAI log filtering. • **all**—Logs all packets that match DHCP bindings. • **none**—Does not log packets that match DHCP bindings. • **permit**—Logs packets permitted by DHCP bindings.
Step 3	(Optional) **show running-config dhcp** **Example:** switch(config)# show running-config dhcp	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Verifying the DAI Configuration

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

Command	Purpose
show ip arp inspection	Displays the status of DAI.
show ip arp inspection 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 17: 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.
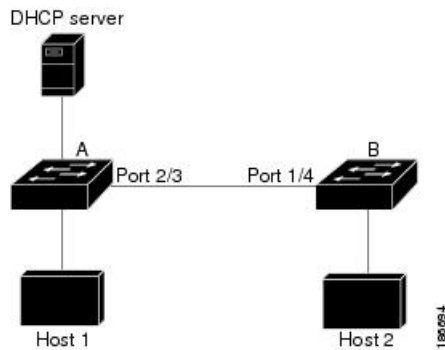
DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically-assigned IP addresses.

- 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#
```

Additional References for DAI

Related Documents

Related Topic	Document Title
ACL TCAM regions	Configuring IP ACLs
DHCP and DHCP snooping	Configuring DHCP

Standards

Standard	Title
RFC-826	An Ethernet Address Resolution Protocol (http://tools.ietf.org/html/rfc826)

Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on 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 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.

Licensing Requirements for IP Source Guard

This table shows the licensing requirements for IP Source Guard.

Product	License Requirement
NX-OS	IP Source Guard 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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 365.

- 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.=

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.=

Default Settings for IP Source Guard

This table lists the default settings for IP Source Guard parameters.

Table 37: 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** **Example:** ``` switch# configure terminal switch(config)# ```	Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface ethernet** *slot*/*port*
	
	Example:
	
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

Step 3	[**no**] **ip verify source dhcp-snooping-vlan**
	
	Example:
	
	`switch(config-if)# ip verify source`
	`dhcp-snooping vlan`
		Enables IP Source Guard on the interface. The **no** form of this command disables IP Source Guard on the interface.

Step 4	(Optional) **show running-config dhcp**
	
	Example:
	
	`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**
	
	Example:
	
	`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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip source binding** *ip-address mac-address* **vlan** *vlan-id* **interface** *interface-type slot*/*port*
	
	Example:
	
	`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*]
	
	Example:
	
	`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.

	Command or Action	Purpose

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

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

Step 2	[**no**] **ip dhcp snooping ipsg-excluded vlan** *vlan-list*
	
	Example:
	`switch(config)# ip dhcp snooping`
	`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*] **Example:** `switch(config)# show ip ver source`	Displays which VLANs are excluded.

Step 4	(Optional) **copy running-config startup-config**
	
	Example:
	`switch(config)# copy running-config`
	`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
DHCP and DHCP snooping	Configuring DHCP

C H A P T E R **19**

Configuring Password Encryption

This chapter describes how to configure password encryption on NX-OS devices.

This chapter includes the following sections:

- About AES Password Encryption and Master Encryption Keys
- Licensing Requirements for Password Encryption
- Guidelines and Limitations for Password Encryption
- Default Settings for Password Encryption
- Configuring Password Encryption
- Verifying the Password Encryption Configuration
- Configuration Examples for Password Encryption=

About AES Password Encryption and Master 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 master encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a master 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 NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Related Topics

Configuring a Master Key and Enabling the AES Password Encryption Feature

Configuring Global RADIUS Keys

Configuring a Key for a Specific RADIUS Server

Configuring Global TACACS+ Keys

Configuring a Key for a Specific TACACS+ Server

Configuring a Master Key and Enabling the AES Password Encryption Feature

Licensing Requirements for Password Encryption

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Password encryption 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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 master keys.

- RADIUS and TACACS+ are the only applications that can use the AES password encryption feature.

- Configurations containing type-6 encrypted passwords are not rollback compliant.

- You can enable the AES password encryption feature without a master key, but encryption starts only when a master key is present in the system.

- Deleting the master key stops type-6 encryption and causes all existing type-6 encrypted passwords to become unusable, unless the same master 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 master key on the device to which the configuration will be applied.

Default Settings for Password Encryption

This table lists the default settings for password encryption parameters.

Table 38: Default Password Encryption Parameter Settings

Parameters	Default
AES password encryption feature	Disabled
Master key	Not configured

Configuring Password Encryption

This section describes the tasks for configuring password encryption on NX-OS devices.

Configuring a Master Key and Enabling the AES Password Encryption Feature

You can configure a master key for type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Procedure

	Command or Action	**Purpose**

Step 1	[**no**] **key config-key ascii**
	
	Example:
	
	`switch# key config-key ascii`
	`New Master Key:`
	`Retype Master Key:`
		Configures a master key to be used with the AES password encryption feature. The master key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the master key at any time.
		
		If you enable the AES password encryption feature before configuring a master key, a message appears stating that password encryption will not take place unless a master key is configured. If a master key is already configured, you are prompted to enter the current master key before entering a new master key.

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

Step 3	[**no**] **feature password encryption aes**
	
	Example:
	
	`switch(config)# feature password`
	`encryption aes`
		Enables or disables the AES password encryption feature.

Step 4	(Optional) **show encryption service stat**
	
	Example:
	
	`switch(config)# show encryption service`
	`stat`
		Displays the configuration status of the AES password encryption feature and the master key.

Step 5	Required: **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.
		
		Note This command is necessary to synchronize the master 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 master key.

Procedure

	Command or Action	Purpose
Step 1	**encryption re-encrypt obfuscated** **Example:** `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.

Before you begin

Ensure that you have configured a master key.

Procedure

	Command or Action	Purpose

Step 1	**encryption decrypt type6**
	Example:
	`switch# encryption decrypt type6`
	`Please enter current Master Key:`
		Converts type-6 encrypted passwords back to their original states.

Deleting Type-6 Encrypted Passwords

You can delete all type-6 encrypted passwords from the NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**encryption delete type6** **Example:** `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 master key.

Configuration Examples for Password Encryption

The following example shows how to create a master 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
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 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 *CN93240YC-FX2 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.

Licensing Requirements for Keychain Management

This table shows the licensing requirements for keychain management.

Product	License Requirement
NX-OS	Keychain management 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

Prerequisites for Keychain Management

Keychain management has no prerequisites.

Guidelines and Limitations for Keychain Management

Keychain management has the following configuration guideline and limitation:

- Changing the system clock impacts when the keys are active.

Default Settings for Keychain Management

This table lists the default settings for NX-OS keychain management parameters.

Table 39: 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**key chain** *name*
	
	Example:
	
	`switch(config)# key chain bgp-keys`
	`switch(config-keychain)#`
		Creates the keychain and enters keychain configuration mode.

Step 3	(Optional) **show key chain** *name*
	
	Example:
	
	`switch(config-keychain)# show key chain`
	` bgp-keys`
		Displays the keychain configuration.

Step 4	(Optional) **copy running-config startup-config** **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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no key chain** *name* **Example:** `switch(config)# no key chain bgp-keys`	Removes the keychain and any keys that the keychain contains.
Step 3	(Optional) **show key chain** *name* **Example:** `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** **Example:** `switch(config-keychain)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Master Key and Enabling the AES Password Encryption Feature

You can configure a master key for type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Procedure

	Command or Action	**Purpose**

Step 1	[**no**] **key config-key ascii**
	
	Example:
	
	`switch# key config-key ascii`
	`New Master Key:`
	`Retype Master Key:`
		Configures a master key to be used with the AES password encryption feature. The master key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the master key at any time.
		
		If you enable the AES password encryption feature before configuring a master key, a message appears stating that password encryption will not take place unless a master key is configured. If a master key is already configured, you are prompted to enter the current master key before entering a new master key.

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

Step 3	[**no**] **feature password encryption aes**
	
	Example:
	
	`switch(config)# feature password`
	`encryption aes`
		Enables or disables the AES password encryption feature.

Step 4	(Optional) **show encryption service stat**
	
	Example:
	
	`switch(config)# show encryption service`
	`stat`
		Displays the configuration status of the AES password encryption feature and the master key.

Step 5	Required: **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.
		
		Note This command is necessary to synchronize the master key in the running configuration and the startup configuration.

Related Topics

About AES Password Encryption and Master Encryption Keys

About AES Password Encryption and Master Encryption Keys

Configuring Text for a Key

Configuring Accept and Send Lifetimes for a Key

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.

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

Step 2	**key chain** *name*
	
	Example:
	`switch(config)# key chain bgp-keys`
	`switch(config-keychain)#`
		Enters keychain configuration mode for the keychain that you specified.

Step 3	**key** *key-ID*
	
	Example:
	`switch(config-keychain)# key 13`
	`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*
	
	Example:
	`switch(config-keychain-key)# key-string`
	` 0 AS3cureStr1ng`
		Configures the text string for the key. The *text-string* argument is alphanumeric, case-sensitive, and supports special characters.
		
		The *encryption-type* argument can be one of the following values:
		
		    • 0—The *text-string* argument that you enter is unencrypted text. This is the default.
		
		    • 7—The *text-string* argument that you enter is encrypted. The encryption method is a 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 NX-OS device.=

Step 5	(Optional) **show key chain** *name* [**mode decrypt**]	Shows the keychain configuration, including the key text configuration. The mode decrypt

	Command or Action	Purpose

	Example:
	
	`switch(config-keychain-key)# show key`
	`chain bgp-keys`
		option, which can be used by a device administrator only, displays the keys in cleartext.

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

Related Topics

Configuring a Master Key and Enabling the AES Password Encryption Feature

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

Step 2	**key chain** *name*
	
	Example:
	
	`switch(config)# key chain bgp-keys`
	`switch(config-keychain)#`
		Enters keychain configuration mode for the keychain that you specified.

Step 3	**key** *key-ID*
	
	Example:
	
	`switch(config-keychain)# key 13`
	`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*] **Example:**	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.

	Command or Action	Purpose
	`switch(config-keychain-key)# accept-lifetime 00:00:00 Jun 13 2013 23:59:59 Sep 12 2013`	The *start-time* argument is the time of day and date that the key becomes active.
		Specify the end of the lifetime with one of the following options:
		• **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
		• **infinite**—The accept lifetime of the key never expires.
		• *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.
Step 5	**send-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*] **Example:** `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.
		The *start-time* argument is the time of day and date that the key becomes active.
		You can specify the end of the send lifetime with one of the following options:
		• **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
		• **infinite**—The send lifetime of the key never expires.
		• *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**] **Example:** `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** **Example:**	Copies the running configuration to the startup configuration.

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

Related Topics

Configuring a Master Key and Enabling the AES Password Encryption Feature

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

Step 2	**key chain** *name*
	
	Example:
	```switch(config)# key chain bgp-keys```
	```switch(config-keychain)#```
		Enters keychain configuration mode for the keychain that you specified.

Step 3	**key** *key-ID*
	
	Example:
	```switch(config-keychain)# key 13```
	```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.
		
		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}**
	
	Example:
	```switch(config-keychain-key)#```
	```cryptographic-algorithm HMAC-SHA-1```
		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*
	
	Example:
	```switch(config-keychain-key)# show key```
	```chain bgp-keys```
		Shows the keychain configuration.

Step 6	(Optional) **copy running-config startup-config**
	
	Example:
	```switch(config-keychain-key)# copy```
	```running-config startup-config```
		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 bgp keys. Each key text string is encrypted. Each key has longer accept lifetimes than send lifetimes, to help prevent lost communications by accidentally configuring a time in which there are no active keys.

```
key chain bgp-keys
  key 0
    key-string 7 zqdest
    accept-lifetime 00:00:00 Jun 01 2013 23:59:59 Sep 12 2013
    send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
  key 1
    key-string 7 uaeqdyito
    accept-lifetime 00:00:00 Aug 12 2013 23:59:59 May 12 2013
    send-lifetime 00:00:00 Sep 12 2013 23:59:59 Aug 12 2013
  key 2
    key-string 7 eekgsdyd
    accept-lifetime 00:00:00 Nov 12 2013 23:59:59 Mar 12 2013
    send-lifetime 00:00:00 Dec 12 2013 23:59:59 Feb 12 2013
```

Where to Go Next

For information about routing features that use keychains, see the *CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*.

Additional References for Keychain Management

Related Documents

Related Topic	Document Title
Border Gateway Protocol	*CN93240YC-FX2 NX-OS Unicast Routing Configuration Guide*
OSPFv2	*CN93240YC-FX2 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.	—

**Inspur-Cisco Networking
Technology Co.,Ltd.**

Configuring Unicast RPF

This chapter describes how to configure unicast reverse path forwarding (uRPF) on NX-OS devices.

This chapter includes the following sections:

- About Unicast RPF
- Licensing Requirements for Unicast RPF
- Guidelines and Limitations for Unicast RPF
- Default Settings for Unicast RPF
- Configuration Examples for Unicast RPF
- Verifying the Unicast RPF Configuration
- Additional References for Unicast RPF=

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 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.

Licensing Requirements for Unicast RPF

Product	License Requirement
NX-OS	Unicast RPF 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 a complete explanation of the NX-OS licensing scheme, see the NX-OS Licensing Guide.

Guidelines and Limitations for Unicast RPF

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

- You must apply uRPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.

- The further downstream that you apply uRPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying uRPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying uRPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying uRPF across many sites does add to the administration cost of operating the network.

- The more entities that deploy uRPF across Internet, intranet, and extranet resources means the better the chances of mitigating large-scale network disruptions throughout the Internet community and of tracing the source of an attack.

- uRPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure uRPF at a home gateway so that uRPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

- You can use uRPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.

- Do not use uRPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure uRPF only where there is natural or configured symmetry.

- 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.

- For CN93240YC-FX2 switches, the ping to a directly connected peer IP interface will fail when the peer IP interface has strict unicast RPF enabled with the ARP/ND to SRC IP is not resolved.

- For CN93240YC-FX2 switches, packets from a host on the interface subnet will drop due to the unicast RPF failure when the ARP/ND for the host IP is not resolved.=

uRPF Configuration		Applied Traffic Check on Source IP Address			
IPv4	IPv6	IP Unipath	IP ECMP	MPLS Encap/VPN/ECMP	Unipath MPLS -
Disable	Disable	Allow	Allow	Allow	Allow
Loose	Loose	uRPF loose	uRPF loose	uRPF loose	uRPF strict

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

- Strict uRPF does not block the ICMP traffic destined to the interface.

Default Settings for Unicast RPF

This table lists the default settings for unicast RPF parameters.

Table 40: Default Unicast RPF Parameter Settings

Parameters	Default
Unicast RPF	Disabled

Configuring Unicast RPF for CN93240YC-FX2 Switches

You can configure one of the following Unicast RPF modes on an ingress interface for CN93240YC-FX2 running 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **system urpf disable**	Enables Unicast RPF on the switch.

	Command or Action	Purpose
	Example: `switch(config)# no system urpf disable`	**Note** You must reload the NX-OS box to apply the Unicast RPF configuration.

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

Step 4	{**ip** | **ipv6**} **address** *ip-address/length*
	Example:
	`switch(config-if)# ip address`
	`172.23.231.240/23`
		Specifies an IPv4 or IPv6 address for the interface.

Step 5	{**ip** | **ipv6**} **verify unicast source reachable-via** {**any** [**allow-default**] | **rx**} **Example:** `switch(config-if)# ip verify unicast source reachable-via any`	Configures Unicast RPF on the interface for both IPv4 and IPv6. You can enable IPv4 and IPv6 uRPF separately for the CN93240YC-FX2 switches. **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.

	Command or Action	Purpose
	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**
	
	Example:
	
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 7	(Optional) **show ip interface ethernet** *slot*/*port* **Example:** `switch(config)# show ip interface` `ethernet 1/54	grep -i "unicast reverse` ` path forwarding"` `IP unicast reverse path forwarding: none`	Displays the IP information for an interface and verifies if Unicast RPF is enabled.

Step 8	(Optional) **show running-config interface ethernet** *slot*/*port*
	
	Example:
	
	`switch(config)# show running-config`
	`interface ethernet 2/3`
		Displays the configuration for an interface in the running configuration.

Step 9	(Optional) **copy running-config startup-config**
	
	Example:
	
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuration Examples for Unicast RPF

The following example shows how to configure loose unicast RPF for IPv4 packets on a CN93240YC-FX2 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 CN93240YC-FX2 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 CN93240YC-FX2 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 CN93240YC-FX2 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	CN93240YC-FX2 NX-API REST SDK User Guide and API Reference
MPLS VPN	CN93240YC-FX2 NX-OS Label Switching Configuration Guide

Configuring Switchport Blocking

This chapter describes how to configure switchport blocking on the NX-OS device.

This chapter includes the following sections:

- About Switchport Blocking
- Licensing Requirements for Switchport Blocking
- Guidelines and Limitations for Switchport Blocking
- Default Settings for Switchport Blocking
- Configuring Switchport Blocking
- Verifying the Switchport Blocking Configuration
- Configuration Example for Switchport Blocking=

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.

Licensing Requirements for Switchport Blocking

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Switchport blocking 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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.
  =

Default Settings for Switchport Blocking

This table lists the default settings for switchport blocking parameters.

Table 41: 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**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

Step 3	[**no**] **switchport block** {**multicast** | **unicast**}
	
	Example:
	
	`switch(config-if)# switchport block`
	`unicast`
		Prevents the flooding of unknown multicast or unicast packets on the specified interface.
		
		Use the **no** form of this command to resume normal forwarding on the port.

	Command or Action	Purpose
Step 4	(Optional) **show interface** [**ethernet** *slot*/*port* | **port-channel** *number*] **switchport** **Example:** `switch(config-if)# show interface ethernet 1/1 switchport`	Displays the switchport blocking configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the 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 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 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 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 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 NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the 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 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 NX-OS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set. You configure packet classifications and rate controlling policies using class maps and policy maps.

Rate Controlling Mechanisms

Once the packets are classified, the 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 *CN93240YC-FX2 NX-OS Quality of Service Configuration Guide*.

Dynamic and Static CoPP ACLs

CoPP access control lists (ACLs) are classified as either dynamic or static.

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.

Default Policing Policies

When you bring up your NX-OS device for the first time, the 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 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

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
```

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 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

Configuring IP ACLs

Configuring MAC ACLs

Licensing Requirements for CoPP

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	CoPP 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 NX-OS licensing scheme, see the *NX-OS Licensing Guide*.

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 based on the data center and application requirements.

- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features used in your specific environment as well as the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.

- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.

- All the traffic that you do not specify in the other class maps is put into the last class, the default class. Monitor the drops in this class and investigate if these drops are based on traffic that you do not want or the result of a feature that was not configured and you need to add.

- All broadcast traffic is sent through CoPP logic in order to determine which packets (for example, ARP and DHCP) need to be redirected through an access control list (ACL) to the router processor. Broadcast traffic that does not need to be redirected is matched against the CoPP logic, and both conforming and violated packets are counted in the hardware but not sent to the CPU. Broadcast traffic that needs to be sent to the CPU and broadcast traffic that does not need to be sent to the CPU must be separated into different classes.

- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.

- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the device. Filtering this traffic could prevent remote access to the NX-OS device and require a console connection.=

- The 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 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 NX-OS release that supports the CoPP feature to a NX-OS release that supports the CoPP feature with additional classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available.

- Before you downgrade from a NX-OS release that supports the CoPP feature to an earlier 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 NX-OS initial setup utility because using it can impact the control plane of the network.=

- The protocol ACL filtering feature has the following limitations:

    - Once the dynamic 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 dynamic CoPP ACLs.

    - You cannot override the existing dynamic CoPP with a new policy. You must remove the existing dynamic 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, which is either 128 MAC entries or 128 IPv4 entries. The device will automatically apply 128 entries for egress CoPP when you carve TCAM for 256 entries.

    - Policer actions are not supported.

    - SNMP MIB support is not required.

    - IPv6 ACL not supported for dynamic CoPP=

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

Default Settings for CoPP

This table lists the default settings for CoPP parameters.

Table 42: Default CoPP Parameters Settings

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

Configuring CoPP

This section describes how to configure CoPP.

Configuring a Control Plane Class Map

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

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

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

Before you begin

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

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name* **Example:** switch(config)# class-map type control-plane ClassMapA switch(config-cmap)#	Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive. **Note** You cannot use class-default, match-all, or match-any as class map names.
Step 3	(Optional) **match access-group name** *access-list-name* **Example:** switch(config-cmap)# match access-group name MyAccessList	Specifies matching for an IP ACL. **Note** The permit and deny ACL keywords are ignored in the CoPP matching.
Step 4	(Optional) **match exception** {**ip** | **ipv6**} **icmp redirect** **Example:** switch(config-cmap)# match exception ip icmp redirect	Specifies matching for IPv4 or IPv6 ICMP redirect exception packets.

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

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

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

Configuring the Control Plane Service Policy

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

✎

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

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

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

	Command or Action	**Purpose**

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

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

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

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

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

Related Topics

[Configuring a Control Plane Policy Map](#)

Configuring the CoPP Scale Factor Per Line Card

You can configure the CoPP scale factor per line card.

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

Procedure

	Command or Action	Purpose

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

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

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

	Command or Action	Purpose
	the **scale-factor 1 module** *multiple-module-range* command.	

Step 4	(Optional) **show policy-map interface control-plane**
	
	Example:
	`switch(config-cp)# show policy-map`
	`interface control-plane`
		Displays the applied scale factor values when a CoPP policy is applied.

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

Changing or Reapplying the Default CoPP Policy

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

Procedure

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

Related Topics

Changing or Reapplying the Default CoPP Policy Using the Setup Utility

Copying the CoPP Best Practice Policy

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

Procedure

	Command or Action	**Purpose**
Step 1	**copp copy profile** {**strict**	**moderate**	**lenient**	**dense**} {**prefix**	**suffix**} *string* **Example:** `switch# copp copy profile strict prefix abc`	Creates a copy of the CoPP best practice policy. CoPP renames all class maps and policy maps with the specified prefix or suffix.
Step 2	(Optional) **show copp status** **Example:** `switch# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the copied policy is not attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch# show running-config copp`	Displays the CoPP configuration in the running configuration, including the copied policy configuration.

Configuring ARP ACL Filtering for CoPP

You can configure MAC ACL filtering at CoPP.

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	[no] hardware access-list tcam region erg-copp *size*
	
	Example:
	`switch(config)# hardware access-list`
	`tcam region erg-copp 256`
		Configures the size of the CoPP TCAM region.

Step 3	copy running-config startup-config
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Step 4	reload **Example:** `switch(config)# reload`	Reloads the device. **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
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 6	mac access-list mac-foo-1
	
	Example:
	`switch# mac access-list mac-foo-1`
	`switch(config-mac-acl)#`	

Step 7	class-map type control-plane [match-all | match-any] *class-map-name*
	
	Example:
	`switch(config)# class-map type`
	`control-plane match-any c-map2`
	`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*
	
	Example:
	`switch(config-cmap)# match access-group`
	`name IP-foo-1`	

Step 9	policy-map type control-plane *policy-map-name* **Example:**	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**

	`switch(config)# policy-map type`
	`control-plane ClassMapA`
	`switch(config-pmap)#`	

Step 10	**class** {*class-map-name* [**insert-before** *class-map-name2*]	**class-default**}
		
		Example:
		`switch(config-pmap)# class ClassMap2`
		`switch(config-pmap-c)#`
			Specifies a control plane class map name or the class default and enters control plane class configuration mode.
			
			The class-default class map is always at the end of the class map list for a policy map.

Step 11	Enter one of the following commands: • **police** [**cir**] {*cir-rate* [*rate-type*]} • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000 packets`	Specifies the committed information rate (CIR). The rate range is as follows: The committed burst (BC) range is as follows:

Step 12	**control-plane Dynamic mode**
	
	Example:
	`switch(config)# control-plane dynamic mode`
	`switch(config-cp-dyn)#`
		Enters the control plane dynamic configuration mode.

Step 13	**service-policy-dynamic input** *policy-map-name*
	
	Example:
	`switch(config-cp-dyn)#`
	`service-policy-dynamic input PolicyMap1`
		Specifies a policy map for the input traffic.

Configuring IP ACL Filtering for 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	[**no**] **hardware access-list tcam region erg-copp** *size* **Example:** `switch(config)# hardware access-list tcam region erg-copp 256`	Configures the size of the egress CoPP TCAM region.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device. **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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 6	**ip access-list IP-foo-1**
	
	Example:
	`switch# ip access-list mac-foo-1`
	`switch(config-acl)#`	

Step 7	**permit tcp access-list IP-foo-1** *eq bgp* **Example:** `switch(config-acl)# 10 permit tcp 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*
	
	Example:
	`switch(config)# class-map type control-plane match-any c-map2`
	`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	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name IP-foo-1`	
Step 10	**policy-map type control-plane** *policy-map-name*	Specifies a control plane policy map and enters policy map configuration mode. The policy

	Command or Action	Purpose

	Example:
	`switch(config)# policy-map type`
	`control-plane ClassMapA`
	`switch(config-pmap)#`
		map name can have a maximum of 64 characters and is case sensitive.

Step 11	**class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
	
	Example:
	`switch(config-pmap)# class ClassMap2`
	`switch(config-pmap-c)#`
		Specifies a control plane class map name or the class default and enters control plane class configuration mode.
		
		The class-default class map is always at the end of the class map list for a policy map.

Step 12	Enter one of the following commands:
	
	　• **police** [**cir**] {*cir-rate* [*rate-type*]}
	　• **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
	　• **police** [**cir**] {*cir-rate* [*rate-type*]]}
	　　conform transmit [**violate drop**]
	
	Example:
	`switch(config-pmap-c)# police cir 52000`
	` bc 1000 packets`
	
	Example:
	`switch(config-pmap-c)# police cir 3400`
	` kbps bc 200 kbytes`
		Specifies the committed information rate (CIR). The rate range is as follows:
		
		The committed burst (BC) range is as follows:

Step 13	**control-plane Dynamic mode**
	
	Example:
	`switch(config)# control-plane dynamic`
	`mode`
	`switch(config-cp-dyn)#`
		Enters the control plane dynamic configuration mode.

Step 14	**service-policy-dynamic input** *policy-map-name*
	
	Example:
	`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.
	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. When you do not include the prior-ver option, this command displays the difference between two currently applied default CoPP best practice policies (such as the currently applied strict and currently applied moderate policies). When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies).
show copp profile {**strict** | **moderate** | **lenient** | **dense**}	Displays the details of the CoPP best practice policy, along with the classes and policer values.

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 CN Release 9.2(3), CoPP supports the 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

Clearing the CoPP Statistics

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

Configuration Examples for CoPP

This section includes example CoPP configurations.

CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-p-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-p-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-p-acl-arp
permit any any 0x0806

ip access-list copp-system-p-acl-tacas
permit udp any any eq 49

ip access-list copp-system-p-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-p-acl-icmp
permit icmp any any

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

class-map type control-plane match-any copp-system-p-class-normal
match access-group name copp-system-p-acl-icmp
match exception ip icmp redirect
```

```
match exception ip icmp unreachable
match exception ip option

policy-map type control-plane copp-system-p-policy

class copp-system-p-class-critical
police cir 19000 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-important
police cir 500 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-normal
police cir 300 pps bc 32 packets conform transmit violate drop

class class-default
police cir 50 pps bc 32 packets conform transmit violate drop

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

Create CoPP class and associate ACL:

```
class-map type control-plane copp-arp-class
match access-group name copp-arp-acl
```

Add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
class copp-arp-class
police pps 500
```

Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup

        ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.


*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.


Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no)[y]: <CR>

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n
```

```
      Enter the switch name : <CR>

      Enable license grace period? (yes/no) [n]: n

      Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

      Configure the default gateway? (yes/no) [y]: n

      Configure advanced IP options? (yes/no) [n]: <CR>

      Enable the telnet service? (yes/no) [n]: y

      Enable the ssh service? (yes/no) [y]: <CR>

        Type of ssh key you would like to generate (dsa/rsa) : <CR>

      Configure the ntp server? (yes/no) [n]: n

      Configure default interface layer (L3/L2) [L3]: <CR>

      Configure default switchport interface state (shut/noshut) [shut]: <CR>

      Configure best practices CoPP profile (strict/moderate/lenient/dense/skip) [strict]:
    strict


    The following configuration will be applied:
      password strength-check
      no license grace-period
      no telnet server enable
      no system default switchport
      system default switchport shutdown
      policy-map type control-plane copp-system-p-policy

    Would you like to edit the configuration? (yes/no) [n]: <CR>

    Use this configuration and save it? (yes/no) [y]: y

    switch#
```

Additional References for CoPP

This section provides additional information related to implementing CoPP.

Related Documents

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

Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker

Configuring Rate Limits

This chapter describes how to configure rate limits for supervisor-bound traffic on 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 NX-OS device. You can configure rate limits in packets per second 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 this option only, you can configure rate limits in kilobits per second.

Licensing Requirements for Rate Limits

The following table shows the licensing requirements for this feature:

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

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).

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

Default Settings for Rate Limits

This table lists the default settings for rate limits parameters.

Table 43: 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	1000 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

Configuring Rate Limits

You can set rate limits on supervisor-bound traffic.

Procedure

	Command or Action	Purpose

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

Step 2	**hardware rate-limiter access-list-log** {*packets* | **disable**} [**module** *module* [**port** *start end*]]
	
	Example:
	
	`switch(config)# hardware rate-limiter`
	`access-list-log 200`
		Configures rate limits in packets per second for packets copied to the supervisor module for access list logging. The range is from 0 to 10000.

	Command or Action	Purpose
Step 3	**hardware rate-limiter bfd** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter bfd 500`	Configures rate limits in packets per second for bidirectional forwarding detection (BFD) packets. The range is from 0 to 10000.
Step 4	**hardware rate-limiter exception** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter exception 500`	Configures rate limits in packets per second for any exception traffic in the system that is not classified by the Control Plane Policing (CoPP) policy. The range is from 0 to 10000.
Step 5	**hardware rate-limiter fex** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter fex 500`	Configures rate limits in packets per second for supervisor-bound FEX trafffic. The range is from 0 to 10000.
Step 6	**hardware rate-limiter layer-3 glean** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter layer-3 glean 500`	Configures rate limits in packets per second for Layer 3 glean packets. The range is from 0 to 10000. 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. **Note** The CoPP policy controls the rate of glean packets that are forwarded due to global punt adjacency, and this rate limiter controls the destination-specific glean packets.
Step 7	**hardware rate-limiter layer-3 multicast local-groups** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter layer-3 multicast local-groups 300`	Configures rate limits in packets per second for Layer 3 multicast data packets that are punted for initiating a shortest-path tree (SPT) join. The range is from 0 to 10000.

	Command or Action	Purpose

Step 8	**hardware rate-limiter span-egress** *rate* [**module** *module*]
	
	Example:
	`switch(config)# hardware rate-limiter`
	`span-egress 123`
		Configures rate limits in kilobits per second for SPAN for egress traffic. The range is from 0 to 100000000.
		
		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** | **span-egress** |**module** *module*] **Example:** `switch# show hardware rate-limiter`	Displays the rate limit configuration. The module range is from 1 to 30.

Step 10	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Monitoring Rate Limits

You can monitor rate limits.

Procedure

	Command or Action	Purpose

Step 1	**show hardware rate-limiter** [**access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **span-egress** | **module** *module*]
	
	Example:
	`switch# show hardware rate-limiter`
	`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*] } **Example:** ``` switch# clear hardware rate-limiter 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: packets 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: packets per second (kilo bits per second for span-egress)
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
NX-OS licensing	*NX-OS Licensing Guide*

Inspur-Cisco Networking
Technology Co.,Ltd.

Configuring MACsec

This document describes how to configure MACsec on 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).

MACsec is supported on CN93240YC-FX2 switches.

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.

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.

Licensing Requirements for MACsec

Product	License Requirement
NX-OS	MACsec requires a Security license. For a complete explanation of the NX-OS licensing scheme and how to obtain and apply licenses, see the *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 currently not supported.

- Layer 2 and Layer 3 port channels (no subinterfaces)

- When you disable the MACsec feature immediately after a switch-over, some sessions end up pending while some peers are in a secure state.

- 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 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 CN 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.=

Keychain restrictions:

• You cannot overwrite the octet string for a MACsec key. Instead, you must create a new key or a new keychain.

• A new key in the keychain is configured when you enter **end** or **exit**. The default timeout for editor mode is 6 seconds. If the key is not configured with the key octet string or/and the send lifetime within the 6-second window, incomplete information may be used to bring up the MACsec session and could result in the session being stuck in an Authorization Pending state. If the MACsec sessions are not converged after the configuration is complete, you might be advised to shut/no shut the ports.

• For a given keychain, key activation times should overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.

Fallback restrictions:

• If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and will show as rekeying on the old CA under status. And the MACsec session on the new key on primary PSK will be in init state.

• Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.

• The key ID (CKN) used in the fallback key chain must not match any of the key IDs (CKNs) used in the primary key chain.

• Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec policy restrictions:

• BPDU packets might 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.

Interoperability restrictions:

- MACsec peers must run the same NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. For interoperability between previous releases and NX-OS Release 9.2(1), you must use keys with the AES_256_CMAC cryptographic algorithm.

- For interoperability between previous releases and NX-OS Release 9.2(1), pad the MACsec key with zeros if it is less than 32 octets.

- On any NX-OS box, you can configure only one unique combination of an alternate MAC address and Ethernet type on all interfaces.

- 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.

- 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

- When you attempt to downgrade from NX-OS Release 9.3(1) to a Cisco NX-OS release without per port channel member MACcsec config 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.

Enabling MACsec

Before you can access the MACsec and MKA commands, you must enable the MACsec feature.

Procedure

	Command or Action	Purpose

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

Step 2	**feature macsec** **Example:** `switch(config)# feature macsec`	Enables MACsec and MKA on the device.

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

Disabling MACsec

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

Step 2	**macsec shutdown** **Example:** `switch(config)# macsec shutdown`	Disables the MACsec configuration on the device. The **no** option restores the MACsec feature.
Step 3	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration. This step is required only if you

Command or Action	Purpose
switch(config)# copy running-config startup-config	want to retain the MACsec in the shutdown state after the swtich 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** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(Optional) [**no**] **key-chain macsec-psk no-show** **Example:** switch(config)# key-chain macsec-psk no-show	Hides the encrypted key octet string in the output of the **show running-config** and **show startup-config** 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. **Note** The octet string is also hidden when you save the configuration to a file.
Step 3	**key chain** *name* **macsec** **Example:** switch(config)# key chain 1 macsec switch(config-macseckeychain)#	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.
Step 4	**key** *key-id* **Example:** switch(config-macseckeychain)# key 1000 switch(config-macseckeychain-macseckey)#	Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64. **Note** The key must consist of an even number of characters.

	Command or Action	Purpose
Step 5	**key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**} **Example:** `switch(config-macseckeychain-macseckey)# key-octet-string abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm AES_256_CMAC`	Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command. **Note** MACsec peers must run the same NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. To interoperate between previous releases and NX-OS Release 7.0(3)I7(2) or a later release, you must use keys with the AES_256_CMAC cryptographic algorithm.
Step 6	**send-lifetime** *start-time* **duration** *duration* **Example:** `switch(config-macseckeychain-macseckey)# send-lifetime 00:00:00 Oct 04 2016 duration 100000`	Configures a send lifetime for the key. By default, the device treats the start time as UTC. The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
Step 7	(Optional) **show key chain** *name* **Example:** `switch(config-macseckeychain-macseckey)# show key chain 1`	Displays the keychain configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-macseckeychain-macseckey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring MACsec Fallback Key

Beginning with 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters the global configuration mode.

Step 2	**interface** *name*
	
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.

Step 3	**macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name* **Example:** `switch(config-if)# macsec keychain kc2 policy abc fallback-keychain fb_kc2`	Specifies the fallback keychain to use after a MACsec session failure due to a key/key ID mismatch or a key expiration. The fallback key ID should not match any key ID from a primary keychain. Fallback keychain configuration for each interface can be changed on the corresponding interface, without removing the MACsec configuration, by reissuing the same command with the fallback keychain name changed. **Note** The command must be entered exactly the same as the existing configuration command for the interface, except for the fallback keychain name. See Configuring a MACsec Keychain and Keys.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Make sure that MACsec is enabled.

Procedure

	Command or Action	**Purpose**

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

Step 2	**macsec policy** *name*
	
	Example:
	
	`switch(config)# macsec policy abc`
	`switch(config-macsec-policy)#`
		Creates a MACsec policy.

Step 3	**cipher-suite** *name*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`cipher-suite GCM-AES-256`	

Step 4	**key-server-priority** *number*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`key-server-priority 0`
		Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.

Step 5	**security-policy** *name*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`security-policy should-secure`
		Configures one of the following security policies to define the handling of data and control packets:
		
		• **must-secure**—Packets not carrying MACsec headers will be dropped.
		
		• **should-secure**—Packets not carrying MACsec headers will be permitted. This is the default value.

Step 6	**window-size** *number*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`window-size 512`
		Configures the replay protection window such that the secured interface will not accept any packet that is less than the configured window size. The range is from 0 to 596000000.

Step 7	**sak-expiry-time** *time*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`sak-expiry-time 100`
		Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0.

Step 8	**conf-offset** *name*
	
	Example:
	
	`switch(config-macsec-policy)#`
	`conf-offset CONF-OFFSET-0`
		Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50. This command might be necessary for

	Command or Action	Purpose
		intermediate switches to use packet headers {dmac, smac, etype} like MPLS tags.
Step 9	(Optional) **show macsec policy** **Example:** `switch(config-macsec-policy)# show macsec policy`	Displays the MACsec policy configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-macsec-policy)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Rotating PSKs

Follow this procedure to rotate PSKs when the SAK expiry time is configured for 60 seconds in the MACsec policy.

Procedure

Step 1 Use the **no sak-expiry-time** command to remove the SAK expiry timer from the MACsec policy.

Note You need to remove the SAK expiry timer only for the number of policies in the configuration. You do not need to remove it for each interface. If you have defined only one policy and applied it to all interfaces, you need to remove the SAK expiry timer only from this policy.

Step 2 Wait for 2 minutes.

Step 3 Use the **key** *key-id* command to program the new key under the keychain.

Step 4 Once the session with the new key is secured, use the **no key** *key-id* command to delete the old key.

Step 5 Wait for 2 minutes.

Step 6 Use the **sak-expiry-timer 60** command to add the SAK rekey timer to the MACsec policy.

About Configurable EAPOL Destination and Ethernet Type

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

Step 2	**interface** *name*
	Example:
	`switch(config)# interface ethernet 1/1`
	`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. **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**
	Example:
	`switch(config-macseckeychain-macseckey)#`
	` 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**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *name*
	
	Example:
	`switch(config)# interface ethernet 1/1`
	`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**
	
	Example:
	`switch(config-macseckeychain-macseckey)#`
	` 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
----------------------------- --------------- ---- ------------ -------- --------------
  -------------- -------------
system-default-macsec-policy  GCM-AES-XPN-256 16   148809600    0        should-secure
  pn-rollover   FALSE
tests1                        GCM-AES-XPN-256 16   148809600    0        should-secure
  pn-rollover   FALSE
tests2                        GCM-AES-XPN-256 16   148809600    0        should-secure
  pn-rollover   FALSE
tests3                        GCM-AES-256     16   148809600    0        should-secure
  pn-rollover   FALSE
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is not configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 075e701e1c5a4a5143475e5a527d7c7c706a6c724306170103555a5c57510b051e47080
a05000101005e0e50510f005c4b5f5d0b5b070e234e4d0a1d0112175b5e cryptographic-algorithm
AES_256_CMAC
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

Displaying MACsec Statistics

You can display MACsec statistics using the following commands.

Command	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# show macsec secy statistics interface ethernet 2/2

Interface Rx Statistics:
Unicast Uncontrolled Pkts: 588464
Multicast Uncontrolled Pkts: 2371
Broadcast Uncontrolled Pkts: 0
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 - Rx Drop: N/A (N9K-X9736C-FX not supported)
Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
In-Octets Uncontrolled: 134073396 bytes
In-Octets Controlled: 155064074 bytes
Input rate for Uncontrolled Pkts: 152 pps
Input rate for Uncontrolled Pkts: 276485 bps
Input rate for Controlled Pkts: 30 pps
Input rate for Controlled Pkts: 43045 bps

Interface Tx Statistics:
Unicast Uncontrolled Pkts: 547976
Multicast Uncontrolled Pkts: 1952
Broadcast Uncontrolled Pkts: 0
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 - Rx Drop: N/A (N9K-X9736C-FX not supported)
Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
Out-Octets Uncontrolled: 124811118 bytes
Out-Octets Controlled: 143365326 bytes
Out-Octets Common: 124811118 bytes
Output rate for Uncontrolled Pkts: 141 pps
Output rate for Uncontrolled Pkts: 257371 bps
Output rate for Controlled Pkts: 28 pps
Output rate for Controlled Pkts: 40138 bps

SECY Rx Statistics:
Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
Control Pkts: 2178
Untagged Pkts: N/A (N9K-X9736C-FX not supported)
```

```
No Tag Pkts: 6
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: 1570
Untagged Pkts: N/A (N9K-X9736C-FX not supported)

SAK Rx Statistics for AN [2]:
Unchecked Pkts: 0
Delayed Pkts: 0
Late Pkts: 0
OK Pkts: 116891
Invalid Pkts: 0
Not Valid Pkts: 0
Not-Using-SA Pkts: 0
Unused-SA Pkts: 0
Decrypted In-Octets: 20871733 bytes
Validated In-Octets: 0 bytes

SAK Tx Statistics for AN [2]:
Encrypted Protected Pkts: 109070
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: 19462062 bytes
```

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
```

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>0200000000000000000000000000000000000000000000000000000000000000
</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>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet3/1</ifname>
          <policy>am2</policy>
          <keychain>kc2/0200000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>
```

```
[TRUNCATED FOR READABILITY]

<ROW_mka_summary>
        <ifname>Ethernet3/32</ifname>
        <policy>am2</policy>
        <keychain>kc2/02000000000000000000000000000000000000000000000000000000
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>
```

```
]]>]]>
```

Related Documentation

Related Topic	Document Title
Keychain management	CN93240YC-FX2 NX-OS Security Configuration Guide
System messages	CN93240YC-FX2 NX-OS System Messages References