Send document comments to nexus7k-docfeedback@cisco.com

Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0

March 19, 2008

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

Text Part Number: OL-12914-03

CONTENTS

Send document comments to nexus7k-docfeedback@cisco.com

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0*. The latest version of this document is available at the following Cisco website:
http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/security/configuration/guide/sec_nx-os_config.html

To check for additional information about Cisco NX-OS Release 4.0, see the *Cisco Nexus 7000 Series NX-OS Release Notes, Release 4.0* available at the following Cisco website:
http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/release/notes/401_nx-os_release_note.html

Table 1 summarizes the new and changed features for the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0*, and tells you where they are documented.

Table 1 New and Changed Features for Release 4.0

Feature	Description	Changed in Release	Where Documented
AAA default user role	You can enable a default user role for remote users who do not have a user role on the device.	4.0(3)	Chapter 2, "Configuring AAA"
Password-strength checking	You can configure NX-OS to allow you to assign only strong passwords.	4.0(3)	Chapter 6, "Configuring User Accounts and RBAC"
SGT propagation for Cisco TrustSec	You can disable security group tag (SGT) propagation on Layer 2 Cisco TrustSec interfaces.	4.0(3)	Chapter 9, "Configuring Cisco TrustSec"
Cisco TrustSec manual configuration	You can configure SAP for Cisco TrustSec manual mode to use 802.1X.	4.0(3)	Chapter 9, "Configuring Cisco TrustSec"
Multiple DHCP server support	You can configure up to four DHCP server addresses per Layer 3 Ethernet interface.	4.0(3)	Chapter 14, "Configuring DHCP Snooping"
IPv6 support in control plane class maps	You can configure the policing of IPv6 packets in control plane class maps.	4.0(3)	Chapter 20, "Configuring Control Plane Policing"
Rate limits for port security packets	You can configure rate limits for port security packets.	4.0(3)	Chapter 21, "Configuring Rate Limits"
Telnet for IPv6	You can create a Telnet session using IPv6.	4.0(2)	Chapter 5, "Configuring SSH and Telnet"

Table 1 New and Changed Features for Release 4.0 (continued)

Feature	Description	Changed in Release	Where Documented
CoPP configuration status	You can display control plane policing (CoPP) configuration status information.	4.0(2)	Chapter 20, "Configuring Control Plane Policing"
Default CoPP policy assignment	You can change the default CoPP policy assignment using the **setup** command.	4.0(2)	Chapter 20, "Configuring Control Plane Policing"
Default CoPP policies	The default CoPP policies the you can assign using the **setup** command have changed.	4.0(2)	Chapter 20, "Configuring Control Plane Policing"

Preface

This preface describes the audience, organization, and conventions of the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0*. It also provides information on how to obtain related documentation.

This chapter includes the following sections:

Audience

This publication is for experienced network administrators who configure and maintain NX-OS devices.

Document Organization

This document is organized into the following chapters:

Chapter	Description
New and Changed Information	Describes the new and changed information for the new Cisco NX-OS software releases.
Chapter 1, "Overview"	Describes the security features supported by the NX-OS software.
Chapter 2, "Configuring AAA"	Describes how to configure authentication, authorization, and accounting (AAA) features.
Chapter 3, "Configuring RADIUS"	Describes how to configure the RADIUS security protocol.
Chapter 4, "Configuring TACACS+"	Describes how to configure the TACACS+ security protocol.
Chapter 5, "Configuring SSH and Telnet"	Describes how to configure Secure Shell (SSH) and Telnet.
Chapter 6, "Configuring User Accounts and RBAC"	Describes how to configure user accounts and role-based access control (RBAC).

Chapter	Description
Chapter 7, "Configuring 802.1X"	Describes how to configure 802.1X authentication.
Chapter 8, "Configuring NAC"	Describes how to configure Network Admission Control (NAC).
Chapter 9, "Configuring Cisco TrustSec"	Describes how to configure Cisco TrustSec integrated security.
Chapter 10, "Configuring IP ACLs"	Describes how to configure IP access control lists (ACLs).
Chapter 11, "Configuring MAC ACLs"	Describes how to configure MAC ACLs.
Chapter 12, "Configuring VLAN ACLs"	Describes how to configure VLAN ACLs.
Chapter 13, "Configuring Port Security"	Describes how to configure port security.
Chapter 14, "Configuring DHCP Snooping"	Describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping.
Chapter 15, "Configuring Dynamic ARP Inspection"	Describes how to configure Address Resolution Protocol (ARP) inspection.
Chapter 16, "Configuring IP Source Guard"	Describes how to configure IP Source Guard.
Chapter 17, "Configuring Keychain Management"	Describes how to configure keychain management.
Chapter 18, "Configuring Traffic Storm Control"	Describes how to configure traffic storm control.
Chapter 19, "Configuring Unicast RPF"	Describes how to configure Unicast Reverse Path Forwarding (Unicast RPF).
Chapter 20, "Configuring Control Plane Policing"	Describes how to configure control plane policing on ingress traffic.
Chapter 21, "Configuring Rate Limits"	Describes how to configure rate limits on egress traffic.

Document Conventions

Command descriptions use these conventions:

Convention	Description
boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
{ }	Elements in curly brackets are required.
[ ]	Elements in square brackets are optional.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

`screen font`	Terminal sessions and information that 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.

This document uses the following conventions:

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

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

Related Documentation

The following Cisco NX-OS documents are published on Cisco.com:

Release Notes

Cisco Nexus 7000 Series NX-OS Release Notes, Release 4.0

NX-OS Configuration Guides

Cisco Nexus 7000 Series NX-OS Getting Started with Virtual Device Contexts, Release 4.0

Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Software Upgrade Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS XML Management Interface User Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS System Messages Reference

Cisco Nexus 7000 Series NX-OS MIB Quick Reference

NX-OS Command References

Cisco Nexus 7000 Series NX-OS Command Reference Master Index, Release 4.0

Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Interfaces Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS System Management Command Reference, Release 4.0

Other Software Document

Cisco Nexus 7000 Series NX-OS Troubleshooting Guide, Release 4.0

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 1

Overview

Cisco NX-OS 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 (AAA)

AAA is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

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

For information on configuring AAA, see Chapter 2, "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.

For information on configuring RADIUS, see Chapter 3, "Configuring RADIUS." For information on configuring TACACS+, see Chapter 4, "Configuring TACACS+."

SSH and Telnet

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

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

For information on configuring SSH and Telnet, see Chapter 5, "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.

For information on configuring user accounts and RBAC, see Chapter 6, "Configuring User Accounts and RBAC."

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 an 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.

For information on configuring 802.1X, see Chapter 7, "Configuring 802.1X."

NAC

Network Admission Control (NAC) allows you to check endpoint devices for security compliancy and vulnerability before these devices are allowed access to the network. This security compliancy check is referred to as *posture validation*. Posture validation allows you to prevent the spread of worms, viruses, and other rogue applications across the network.

NAC validates that the posture, or state, of endpoint devices complies with security policies before the devices can access protected areas of the network. For devices that comply with the security policies, NAC allows access to protected services in the network. For devices that do not comply with security policies, NAC restricts access to the network that is sufficient only for remediation, which checks the posture of the device again.

For information on configuring NAC, see Chapter 8, "Configuring NAC."

Cisco TrustSec

The Cisco TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Each device in the cloud is authenticated by its neighbors. Communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and replay protection mechanisms. Cisco TrustSec also uses the device and user identification information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Cisco TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic. Cisco TrustSec uses ingress tagging and egress filtering to enforce access control policy in as a conversation.

For information on configuring NAC, see Chapter 9, "Configuring Cisco TrustSec."

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.

For information on configuring IP ACLs, see Chapter 10, "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.

For information on configuring MAC ACLs, see Chapter 11, "Configuring MAC ACLs."

VACLs

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

For information on configuring VACLs, see Chapter 12, "Configuring VLAN ACLs."

Port Security

Port security allows you to configure Layer 2 interfaces that 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.

For information on configuring port security, see Chapter 13, "Configuring Port Security."

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 also use information stored in the DHCP snooping binding database.

For information on configuring DHCP snooping, see Chapter 14, "Configuring DHCP Snooping."

Dynamic ARP Inspection

Dynamic ARP inspection (DAI) ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, an 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 is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the device. 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.

For information on configuring DAI, see Chapter 15, "Configuring Dynamic ARP Inspection."

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.

For information on configuring IP Source Guard, see Chapter 16, "Configuring IP Source Guard."

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 information on configuring keychain management, see Chapter 17, "Configuring Keychain Management."

Traffic Storm Control

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming traffic over a 1-second interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

For information on configuring traffic storm control, see Chapter 18, "Configuring Traffic Storm Control."

Unicast RPF

The Unicast Reverse Path Forwarding (RPF) feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP 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 IP 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.

For information on configuring control plane policing, see Chapter 19, "Configuring Unicast RPF."

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.

For information on configuring control plane policing, see Chapter 20, "Configuring Control Plane Policing."

Rate Limits

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

For information on configuring rate limits, see Chapter 21, "Configuring Rate Limits."

CHAPTER 2

Configuring AAA

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

This chapter includes the following sections:

Information About AAA

This section includes the following topics:

AAA Security Services

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

Based on the user ID and password combination that you provide, Cisco NX-OS devices perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the 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. Cisco NX-OS devices allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).
- Authorization—Provides access control.

  AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the 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 implement the same AAA protocol. The purpose of a server group is to provide for fail-over servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Cisco NX-OS device encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

AAA configuration in Cisco NX-OS devices is service based, which means that you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication
- Cisco TrustSec authentication (see Chapter 9, "Configuring Cisco TrustSec")
- 802.1X authentication (see Chapter 7, "Configuring 802.1X")
- Extensible Authentication Protocol over User Datagram Protocol (EAPoUDP) authentication for Network Admission Control (NAC) (see Chapter 8, "Configuring NAC")
- User management session accounting
- 802.1X accounting (see Chapter 7, "Configuring 802.1X")

Table 2-1 provides the related CLI command for each AAA service configuration option.

Table 2-1 AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
Cisco TrustSec authentication	**aaa authentication cts default**
802.1X authentication	**aaa authentication dot1x default**
EAPoUDP authentication	**aaa authentication eou default**

Table 2-1 AAA Service Configuration Commands (continued)

AAA Service Configuration Option	Related Command
User session accounting	**aaa accounting default**
802.1X accounting	**aaa accounting dot1x default**

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

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

Note If the method is all RADIUS servers, rather than a specific server group, 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.

Table 2-2 shows the AAA authentication methods that you can configure for the AAA services.

Table 2-2 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
Cisco TrustSec authentication	Server groups only
802.1X authentication	Server groups only
EAPoUDP authentication	Server groups only
User management session accounting	Server groups and local
802.1X accounting	Server groups and local

Note For console login authentication and 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.

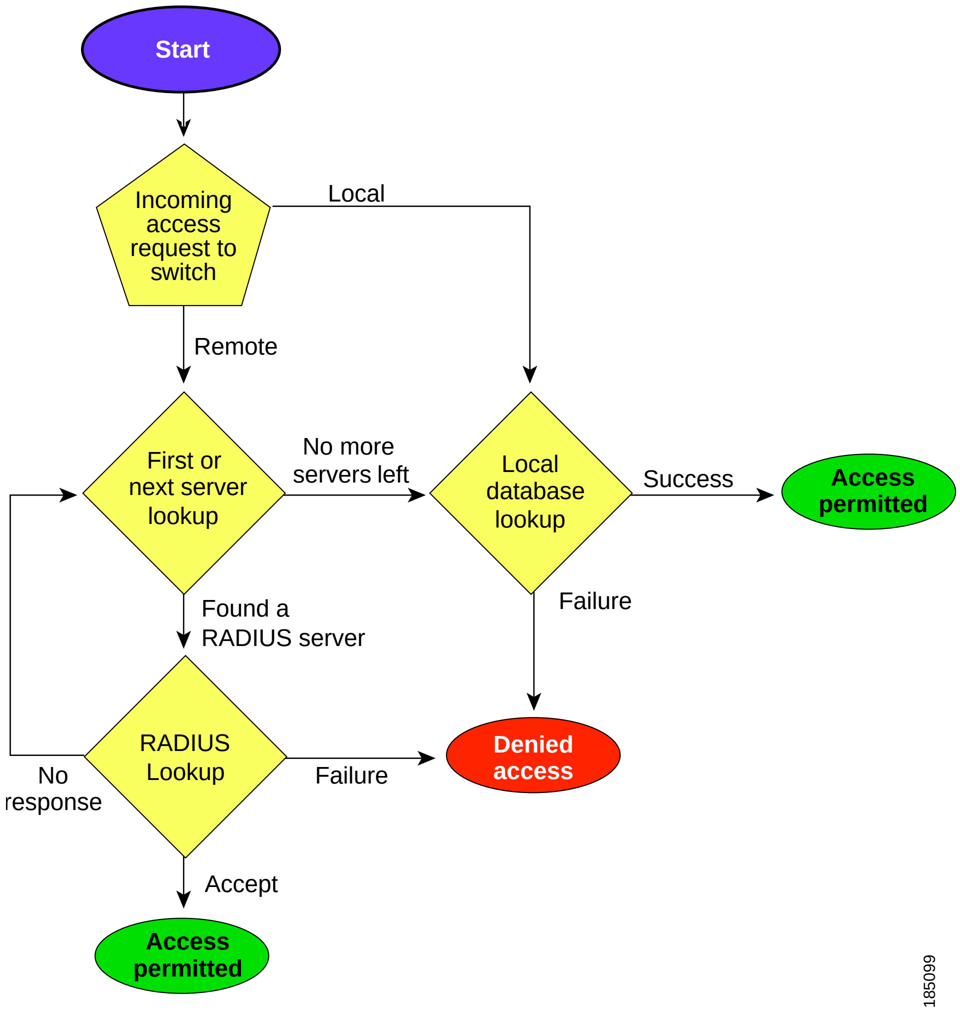
Authentication and Authorization Process for User Login

Figure 2-1 shows a flow chart of the authentication and authorization process for user login. The following list explain the process:

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

2. 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, then the next AAA server is tried and so on until the remote server responds to the authentication request.
   - If all AAA servers in the server group fail to respond, then the servers in the next server group are tried.
   - If all configured methods fail, then the local database is used for authentication.
3. 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 the user roles are not successfully retrieved from the remote AAA server, then the user is assigned with the vdc-operator role.
4. 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.

Figure 2-1 Authorization and Authentication Flow for User Login



> **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.

Virtualization Support

All AAA configuration and operations are local to the VDC, except the default console methods and the AAA accounting log. The configuration and operation of the AAA authentication methods for the console login apply only to the default VDC. The AAA accounting log is only in the default VDC. You can display the contents from any VDC but you must clear it in the default VDC.

For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that at least one RADIUS or TACACS+ server is IP reachable (see the "Configuring RADIUS Server Hosts" section on page 3-6 and the "Configuring TACACS+ Server Hosts" section on page 4-8).
- Ensure that the NX-OS device is configured as a client of the AAA servers.
- Ensure that the preshared 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 (see the "Manually Monitoring RADIUS Servers or Groups" section on page 3-18 and the "Manually Monitoring TACACS+ Servers or Groups" section on page 4-19).

AAA Guidelines and Limitations

RADIUS has the following guidelines and limitations:

- The Cisco NX-OS software does not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally, and does not create local users with all numeric names. If an all numeric username exists on an AAA server and is entered during login, the NX-OS device does log in the user.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

Configuring AAA

This section includes the following topics:

- Process for Configuring AAA, page 2-8
- Configuring Console Login Authentication Methods, page 2-8
- Configuring Default Login Authentication Methods, page 2-10
- Enabling the Default User Role for AAA Authentication, page 2-11
- Enabling Login Authentication Failure Messages, page 2-12

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

Process for Configuring AAA

Follow these steps to configure AAA authentication and accounting:

Step 1 If you want to use remote RADIUS or TACACS+ servers for authentication, configure the hosts on your NX-OS device (see Chapter 3, "Configuring RADIUS" and Chapter 4, "Configuring TACACS+").

Step 2 Configure console login authentication methods (see the "Configuring Console Login Authentication Methods" section on page 2-8).

Step 3 Configure default login authentication methods for user logins (see the "Configuring Default Login Authentication Methods" section on page 2-10).

Step 4 Configure default AAA accounting default methods (see the "Configuring AAA Accounting Default Methods" section on page 2-15).

Note To configure authentication methods for 802.1X, see the "Configuring AAA Authentication Methods for 802.1X" section on page 7-11. To configure authentication methods for EAPoUDP, see the "Enabling the Default AAA Authenication Method for EAPoUDP" section on page 8-16.

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 or TACACS+ servers
- Local database on the NX-OS device
- Username only (**none**)

The default method is local.

Note The configuration and operation of the AAA for the console login apply only to the default VDC

> **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.

BEFORE YOU BEGIN

Ensure that you are in the default VDC.

Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **config t**
2. **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}
3. **exit**
4. **show aaa authentication**
5. **copy running-config start-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`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 TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond.

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

	Command	Purpose
Step 4	**show aaa authentication** **Example:** `switch# show aaa authentication`	(Optional) Displays the configuration of the console login authentication methods.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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 or TACACS+ servers
- Local database on the NX-OS device
- Username only

The default method is local.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **config t**
2. **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}
3. **exit**
4. **show aaa authentication**
5. **copy running-config start-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond.

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

Step 4	**show aaa authentication** **Example:** `switch# show aaa authentication`	(Optional) Displays the configuration of the default login authentication methods.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling the Default User Role for AAA Authentication

In Cisco NX-OS Release 4.0(3) and later releases, you can allow remote users who do not have a user role to log in to the device through RADIUS or TACACS+ using a default user role. You can enable or disable this feature for the VDC as needed. For the default VDC, the default role is network-operator. For nondefault VDCs, the default VDC is vdc-operator. When you disable the AAA default user role feature, remote users who do not have a user role cannot log in to the device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **aaa user default-role**
3. **exit**
4. **show aaa user default-role**
5. **copy running-config start-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show aaa user default-role** **Example:** `switch# show aaa user default-role`	(Optional) Displays the AAA default user role configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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 message is displayed on the user's terminal—if you have enabled displaying login failure messages:

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**

2. **aaa authentication login error-enable**
3. **exit**
4. **show aaa authentication**
5. **copy running-config start-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show aaa authentication** **Example:** `switch# show aaa authentication`	(Optional) Displays the login failure message configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. You can use MSCHAP for user logins to an 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 MSCHAP, you need to configure your RADIUS server to recognize the MSCHAP vendor-specific attributes (VSAs). See the "Using AAA Server VSAs with Cisco NX-OS Devices" section on page 2-16. Table 2-3 shows the RADIUS VSAs required for MSCHAP.

Table 2-3 MSCHAP RADIUS VSAs

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **aaa authentication login mschap enable**
3. **exit**
4. **show aaa authentication login mschap**
5. **copy running-config start-config**

DETAILED STEPS

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

Configuring AAA Accounting Default Methods

Cisco 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **config t**
2. **aaa accounting default** {**group** *group-list* | **local**}
3. **exit**
4. **show aaa accounting**
5. **copy running-config start-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 of the following: • **radius**—Uses the global pool of RADIUS servers for accounting. • *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is **local,** which is used when no server groups are configured or when all the configured server groups fail to respond.

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

Step 4	**show aaa accounting** **Example:** `switch# show aaa accounting`	(Optional) Displays the configuration AAA accounting default methods.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Using AAA Server VSAs with Cisco NX-OS Devices

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

This section includes the following topics:

- About VSAs, page 2-17
- VSA Format, page 2-17
- Specifying Cisco NX-OS User Roles and SMNPv3 Parameters on AAA Servers, page 2-18

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 seperator value *
```

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

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

VSA Format

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

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

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

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space. For example, if you belong to roles network-operator and vdc-admin, the value field would be “network-operator vdc-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 vdc-admin”
  shell:roles*“network-operator vdc-admin”
  ```

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

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

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

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

Specifying Cisco NX-OS User Roles and SMNPv3 Parameters on AAA Servers

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

```
shell:roles="roleA roleB …"
```

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

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

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

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

For more information on user roles, see Chapter 6, "Configuring User Accounts and RBAC."

Displaying and Clearing the Local AAA Accounting Log

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

Note The AAA accounting log is local to the default VDC. You can display the contents from any VDC but you must clear it in the default VDC.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC before clearing the AAA accounting log.

SUMMARY STEPS

1. **show accounting log** [*size*] [**start-time** *year month day hh*:*mm*:*ss*]
2. **clear accounting log**

DETAILED STEPS

	Command	Purpose
Step 1	**show accounting log** [*size*] [**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 start time for the log output.
Step 2	**clear accounting log** **Example:** `switch# clear aaa accounting log`	(Optional) Clears the accounting log contents.

Verifying AAA Configuration

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

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

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example AAA Configuration

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
```

Default Settings

Table 2-4 lists the default settings for AAA parameters.

Table 2-4 Default AAA Parameters

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

Additional References

For additional information related to implementing AAA, see the following sections:

- Related Documents, page 2-20
- Standards, page 2-20
- MIBs, page 2-20

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
RADIUS security protocol	Chapter 3, "Configuring RADIUS"
TACACS+ Security protocol	Chapter 4, "Configuring TACACS+"

Standards

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

MIBs

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

CHAPTER 3

Configuring RADIUS

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

This chapter includes the following sections:

Information About RADIUS

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

This section includes the following topics:

RADIUS Network Environments

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

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

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support authentication profiles. Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the 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 an NX-OS device using RADIUS, the following process occurs:

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

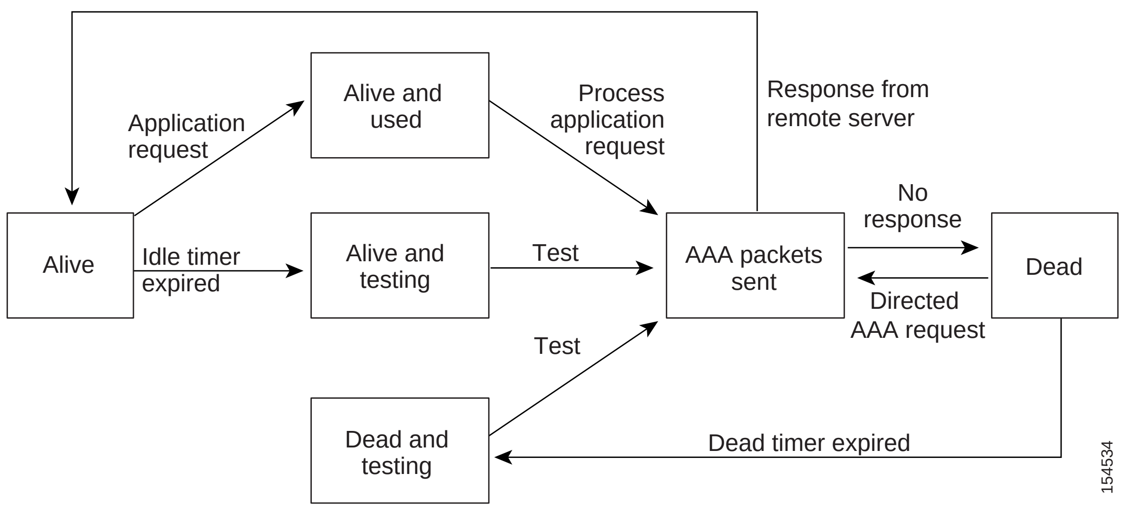
The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

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

RADIUS Server Monitoring

An unresponsive RADIUS server can cause 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. See Figure 3-1.

Figure 3-1 RADIUS Server States



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

Vendor-Specific Attributes

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

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

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

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

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

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

The Cisco NX-OS software supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be "network-operator vdc-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 as supported by Cisco ACS:

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

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

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

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

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

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

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

Virtualization Support

RADIUS configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

The NX-OS device uses virtual routing and forwarding instances (VRFs) to access the RADIUS servers. For more information on VRFs, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*.

Licensing Requirements for RADIUS

The following 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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.
- Obtain preshared keys from the RADIUS servers.
- Ensure that the NX-OS device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations

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

Configuring RADIUS Servers

This section includes the following topics:

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

RADIUS Server Configuration Process

Follow these steps to configure RADIUS servers:

Step 1 Establish the RADIUS server connections to the NX-OS device (see the "Configuring RADIUS Server Hosts" section on page 3-6).

Step 2 Configure the preshared secret keys for the RADIUS servers (see the "Configuring Global Preshared Keys" section on page 3-7).

Step 3 If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods (see the "Allowing Users to Specify a RADIUS Server at Login" section on page 3-11 and the "Configuring AAA" section on page 2-7).

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

- Dead-time interval (see the "Configuring the Dead-Time Interval" section on page 3-17.
- Allow specification of a RADIUS server at login (see the "Allowing Users to Specify a RADIUS Server at Login" section on page 3-11).
- Transmission retry count and timeout interval (see the "Configuring the Global RADIUS Transmission Retry Count and Timeout Interval" section on page 3-12).
- Accounting and authentication attributes (see the "Configuring Accounting and Authentication Attributes for RADIUS Servers" section on page 3-14).

Step 5 If needed, configure periodic RADIUS server monitoring (see the "Configuring Periodic RADIUS Server Monitoring" section on page 3-16).

Configuring RADIUS Server Hosts

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*}
3. **exit**
4. **show radius-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config)# radius-server host 10.10.1.1`	Specifies the IPv4 or IPv6 address or hostname for a RADIUS server.

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

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

Configuring Global Preshared Keys

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Obtain the preshared key values for the remote RADIUS servers.

SUMMARY STEPS

1. **config t**
2. **radius-server key** [**0** | **7**] *key-value*
3. **exit**
4. **show radius-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**radius-server key** [**0** | **7**] *key-value* **Example:** `switch(config)# radius-server key 0 QsEfThUkO`	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.

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

Step 4	**show radius-server** **Example:** `switch# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring RADIUS Server Preshared Keys

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Obtain the preshared key values for the remote RADIUS servers.

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** *key-value*
3. **exit**
4. **show radius-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value* **Example:** `switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.

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

Step 4	**show radius-server** **Example:** `switch# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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 up to 100 server groups in a VDC.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service. For information on AAA services, see the "Remote AAA Services" section on page 2-3.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **aaa group server radius** *group-name*
3. **server** {*ipv4-address* | *ipv6-address* | *server-name*}
4. **deadtime** *minutes*

5. **use-vrf** *vrf-name*
6. **exit**
7. **show radius-server groups** [*group-name*]
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters 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`* | *`server-name`*} **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 4	**`deadtime`** *`minutes`* **Example:** `switch(config-radius)# deadtime 30`	(Optional) 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 (see the "Configuring the Dead-Time Interval" section on page 3-17).
Step 5	**`use-vrf`** *`vrf-name`* **Example:** `switch(config-radius)# use-vrf vrf1`	(Optional) Specifies the VRF to use to contact the servers in the server group.

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

Step 7	**`show radius-server groups`** [*`group-name`*] **Example:** `switch(config)# show radius-server group`	(Optional) Displays the RADIUS server group configuration.
Step 8	**`copy running-config startup-config`** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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 authenticate 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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server directed-request**
3. **exit**
4. **show radius-server directed-request**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)#` **`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	**`exit`**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command	Purpose
Step 4	**show radius-server directed-request** **Example:** `switch# show radius-server directed-request`	(Optional) Displays the directed request configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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 switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server retransmission** *count*
3. **radius-server timeout** *seconds*
4. **exit**
5. **show radius-server**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)#` **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.

	Command	Purpose

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

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

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

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

By default, an 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **retransmit** *count*
3. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*
4. **exit**
5. **show radius-server**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)#` **`radius-server host`** {*ipv4-address* | *ipv6-address* | *host-name*} **`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 in Step 2.
Step 3	`switch(config)#` **`radius-server host`** {*ipv4-address* | *ipv6-address* | *host-name*} **`timeout`** *seconds* **Example:** `switch(config)# radius-server host server1 timeout 10`	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers in Step 3.

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

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*
3. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**

4. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port*
5. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication**
6. **exit**
7. **show radius-server**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.1.1 acct-port 2004`	(Optional) Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting** **Example:** `switch(config)# radius-server host 10.10.1.1 accounting`	(Optional) Specifies that the specified RADIUS server to use only for accounting purposes. The default is both accounting and authentication.
Step 4	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.2.2 auth-port 2005`	(Optional) 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	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication** **Example:** `switch(config)# radius-server host 10.10.2.2 authentication`	(Optional) Specifies that the specified RADIUS server is to be used only for authentication purposes. The default is both accounting and authentication.

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

Step 7	**show radius-server** **Example:** `switch(config)# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 8	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Periodic RADIUS Server Monitoring

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

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

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. **radius-server dead-time** *minutes*
4. **exit**
5. **show radius-server**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**`radius-server host`** {*ipv4-address* | *ipv6-address* | *host-name*} **`test`** {**`idle-time`** *minutes* | **`password`** *password* [**`idle-time`** *minutes*] | **`username`** *name* [**`password`** *password* [**`idle-time`** *minutes*]]} **Example:** `switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

	Command	Purpose
Step 3	**radius-server dead-time** *minutes* **Example:** `switch(config)# radius-server dead-time 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. The valid range is 1 to 1440 minutes.

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

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

Configuring the Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the 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 (see the "Configuring RADIUS Server Groups" section on page 3-9).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **radius-server deadtime** *minutes*
3. **exit**
4. **show radius-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)#` **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	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Manually Monitoring RADIUS Servers or Groups

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **test aaa server radius** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password*
2. **test aaa group** *group-name username password*

DETAILED STEPS

	Command	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *ipv6-address* | *server-name*} [**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 1	**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.

Verifying RADIUS Configuration

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

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

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying RADIUS Server Statistics

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command	Purpose

Step 1	`switch# show radius-server statistics {hostname | ipv4-address | ipv6-address}`
	
	Example:
	`switch# show radius-server statistics 10.10.1.1`
		Displays the RADIUS statistics.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example RADIUS Configuration

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
```

Where to Go Next

You can now configure AAA authentication methods to include the RADIUS server groups (see Chapter 2, “Configuring AAA”).

Default Settings

Table 3-1 lists the default settings for RADIUS parameters.

Table 3-1 Default RADIUS Parameters

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

Additional References

For additional information related to implementing RADIUS, see the following sections:

- Related Documents, page 3-21
- Standards, page 3-21
- MIBs, page 3-21

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
VRF configuration	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Standards

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

MIBs

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

Send document comments to nexus7k-docfeedback@cisco.com

CHAPTER 4

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:

Information About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to an 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. Cisco NX-OS devices provide centralized authentication using the TACACS+ protocol.

This section includes the following topics:

- TACACS+ Advantages, page 4-2
- TACACS+ Operation for User Login, page 4-2
- Default TACACS+ Server Encryption Type and Preshared Key, page 4-3
- TACACS+ Server Monitoring, page 4-3
- Vendor-Specific Attributes, page 4-4
- Virtualization Support, page 4-5

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 an NX-OS device using TACACS+, the following actions occur:

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

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

2. The NX-OS device will eventually receive one of the following responses from the TACACS+ daemon:
   a. ACCEPT—User authentication succeeds and service begins. If the NX-OS device requires user authorization, authorization begins.
   b. REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   c. ERROR—An error occurred at some time during authentication either at the daemon or in the network connection 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 Preshared Key

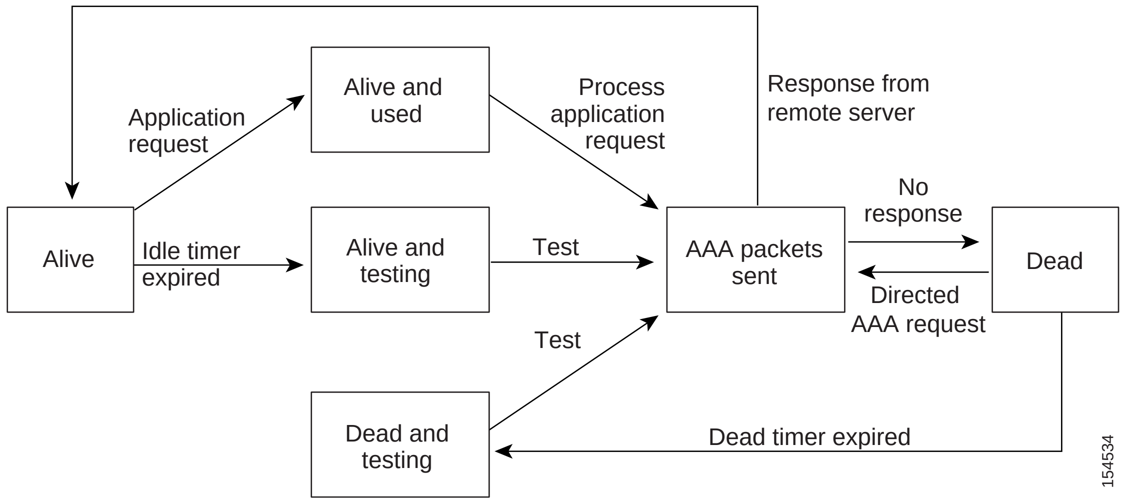
You must configure the TACACS+ preshared key to authenticate the switch to the TACACS+ server. A preshared key is a secret text string shared between the NX-OS device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white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

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

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. An NX-OS device can periodically monitor an 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. An 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 an 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. See Figure 4-1.

Figure 4-1 TACACS+ Server States



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

Vendor-Specific Attributes

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

This section includes the following topics:

- Cisco VSA Format, page 4-4
- Cisco TACACS+ Privilege Levels, page 4-5

Cisco VSA Format

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

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

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

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

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

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

The Cisco NX-OS software supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be "network-operator vdc-admin." This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Cisco ACS:

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

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

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

- accountinginfo—Stores accounting information in addition to the attributes covered by a standard 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).

Cisco TACACS+ Privilege Levels

TACACS+ servers support privilege levels for specifying the permissions that users have when logging into an NX-OS device. For the maximum privilege level 15, the Cisco NX-OS software applies the network-admin role in the default VDC or the vdc-admin role for nondefault VDCs. All other privilege levels are translated to the vdc-operator role. For more information on user roles, see Chapter 6, "Configuring User Accounts and RBAC."

Note If you specify a user role in the cisco-av-pair, that takes precedence over the privilege level.

Virtualization Support

TACACS+ configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

The NX-OS device uses virtual routing and forwarding instances (VRFs) to access the TACACS+ servers. For more information on VRFs, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.
- Obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the NX-OS device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations

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

Configuring TACACS+

This section includes the following topics:

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

TACACS+ Server Configuration Process

To configure TACACS+ servers, follow these steps:

Step 1 Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

Step 2 Establish the TACACS+ server connections to the NX-OS device (see the "Configuring TACACS+ Server Hosts" section on page 4-8).

Step 3 Configure the preshared secret keys for the TACACS+ servers (see the "Configuring Global Preshared Keys" section on page 4-9 and the "Configuring TACACS+ Server Preshared Keys" section on page 4-10).

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods (see the "Configuring TACACS+ Server Groups" section on page 4-11 and the "Configuring AAA" section on page 2-7).

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

- Dead-time interval (see the "Configuring the Dead-Time Interval" section on page 4-18
- TACACS+ server specification allowed at user login (see the "Specifying a TACACS+ Server at Login" section on page 4-13).
- Timeout interval (see the "Configuring the Global TACACS+ Timeout Interval" section on page 4-14).
- TCP port (see the "Configuring TCP Ports" section on page 4-16).

Step 6 If needed, configure periodic TACACS+ server monitoring (see the "Configuring Periodic TACACS+ Server Monitoring" section on page 4-17).

Enabling TACACS+

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **feature tacacs+**

3. **exit**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters 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	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

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

If a preshared key is not configured for a configured TACACS+ server, a warning message is issued if a global key is not configured. If a TACACS+ server key is not configured, the global key (if configured) is used for that server (see the "Configuring Global Preshared Keys" section on page 4-9 and the "Configuring TACACS+ Server Preshared Keys" section on page 4-10).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

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

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*}
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config)# tacacs-server host 10.10.2.2`	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.

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

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

Configuring Global Preshared Keys

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

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

SUMMARY STEPS

1. **config t**
2. **tacacs-server key** [**0** | **7**] *key-value*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server key** [**0** | **7**] *key-value* **Example:** `switch(config)# tacacs-server key 0 QsEfThUkO`	Specifies a preshared key for all TACACS+ servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.

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

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

Configuring TACACS+ Server Preshared Keys

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the “Enabling TACACS+” section on page 4-7).

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

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** *key-value*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value* **Example:** `switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.

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

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

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. For information on AAA services, see the "Remote AAA Services" section on page 2-3.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **aaa group server tacacs+** *group-name*
3. **server** {*ipv4-address* | *ipv6-address* | *host-name*}

4. **deadtime** *minutes*
5. **use-vrf** *vrf-name*
6. **exit**
7. **show tacacs-server groups**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters 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* | *host-name*} **Example:** `switch(config-tacacs+)# server 10.10.2.2`	Configures the TACACS+ server as a member of the TACACS+ server group. **Tip** If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.
Step 4	**deadtime** *minutes* **Example:** `switch(config-tacacs+)# deadtime 30`	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value (see the "Configuring the Dead-Time Interval" section on page 4-18).
Step 5	**use-vrf** *vrf-name* **Example:** `switch(config-tacacs+)# use-vrf vrf1`	(Optional) Specifies the virtual routing and forwarding instance (VRF) to use to contact this server group.

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

Step 7	**show tacacs-server groups** **Example:** `switch(config)# show tacacs-server groups`	(Optional) Displays the TACACS+ server group configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Specifying a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request by enabling the directed-request option. By default, a Cisco 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server directed-request**
3. **exit**
4. **show tacacs-server directed-request**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters 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	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Configuring the Global TACACS+ Timeout Interval

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server timeout** *seconds*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server timeout** *seconds* **Example:** `switch(config)# tacacs-server timeout 10`	Specifies the timeout interval for TACACS+ servers. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds.

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

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

Configuring the Timeout Interval for a Server

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **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	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Configuring TCP Ports

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **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	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco 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 TACACS+ database.

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

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. **tacacs-server dead-time** *minutes*
4. **exit**
5. **show tacacs-server**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **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 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 check 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	**show tacacs-server** **Example:** `switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco 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 (see the "Configuring TACACS+ Server Groups" section on page 4-11).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **config t**
2. **tacacs-server deadtime** *minutes*
3. **exit**
4. **show tacacs-server**
5. **copy running-config startup-config**

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters 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

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

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password*
2. **test aaa group** *group-name username password*

Send document comments to nexus7k-docfeedback@cisco.com

DETAILED STEPS

	Command	Purpose
Step 1	**test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *host-name*} [**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 1	**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.

Disabling TACACS+

You can disable TACACS+.

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **no feature tacacs+**
3. **exit**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**no feature tacacs+** **Example:** `switch(config)# no feature tacacs+`	Disables TACACS+.

	Command	Purpose

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

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

Displaying TACACS+ Statistics

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable TACACS+ (see the "Enabling TACACS+" section on page 4-7).

SUMMARY STEPS

1. **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

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

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Verifying TACACS+ Configuration

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

Command	Purpose
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.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example TACACS+ Configurations

The following example shows how to configure TACACS+:

```
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
```

Where to Go Next

You can now configure AAA authentication methods to include the TACACS+ server groups (see Chapter 2, "Configuring AAA").

Default Settings

Table 4-1 lists the default settings for TACACS+ parameters.

Table 4-1 Default TACACS+ Parameters

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes

Table 4-1 Default TACACS+ Parameters (continued)

Parameters	Default
Periodic server monitoring username	test
Periodic server monitoring password	test

Additional References

For additional information related to implementing TACACS+, see the following sections:

- Related Documents, page 4-23
- Standards, page 4-23
- MIBs, page 4-23

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
VRF configuration	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Standards

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

MIBs

MIBs	MIBs Link
• CISCO-AAA-SERVER-MIB • CISCO-AAA-SERVER-EXT-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/enigmatic/cant/mibs.shtml

CHAPTER 5

Configuring SSH and Telnet

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

This chapter includes the following sections:

Information About SSH and Telnet

This section includes the following topics:

SSH Server

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

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

SSH Client

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

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

SSH Server Keys

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

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

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 two types of key-pairs for use by SSH version 2:

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

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

SSH supports the following public key formats:

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

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

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.

Virtualization Support

SSH and Telnet configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for SSH

SSH and Telnet have the following prerequisites:

- 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

SSH and Telnet have the following configuration guidelines and limitations:

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

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

Configuring SSH

This section includes the following sections:

Generating SSH Server Keys

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **no ssh server enable**
3. **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}
4. **ssh server enable**
5. **exit**
6. **show ssh key**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no ssh server enable** **Example:** `switch(config)# no ssh server enable`	Disables SSH.
Step 3	**ssh key** {**dsa** [**force**] | **rsa** [*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 key. The range is from 768 to 2048 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 4	**ssh server enable** **Example:** `switch(config)# ssh server enable`	Enables SSH.

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

	Command	Purpose
Step 6	**show ssh key** **Example:** `switch# show ssh key`	(Optional) Displays the SSH server keys.
Step 7	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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
- Public Key Certificate in PEM format

Specifying the SSH Public Keys in OpenSSH Format

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Generate an SSH public key in OpenSSH format.

SUMMARY STEPS

1. **config t**
2. **username** *username* **sshkey** *ssh-key*
3. **exit**
4. **show user-account**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**username** *username* **sshkey** *ssh-key* **Example:** `switch(config)# username User1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+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	**show user-account** **Example:** `switch# show user-account`	(Optional) Displays the user account configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

You have generated an SSH public key in IETF SCHSH format.

SUMMARY STEPS

1. **copy** *server-file* **bootflash:***filename*
2. **config t**
3. **username** *username* **sshkey file bootflash:***filename*
4. **exit**
5. **show user-account**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	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	**config t**
	
	Example:
	`switch# config t`
	`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.

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

Starting SSH Sessions

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

BEFORE YOU BEGIN

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

Enable the SSH server on the remote device.

SUMMARY STEPS

1. **ssh** [*username***@**]{*hostname* | *username***@***hostname*} [**vrf** *vrf-name*]

   ssh6 [*username***@**]{*hostname* | *username***@***hostname*} [**vrf** *vrf-name*]

DETAILED STEPS

	Command	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.
	ssh6 [*username*@]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh6 HostA`	Creates an SSH IPv6 session to a remote device using IPv6.

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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **clear ssh hosts**

DETAILED STEPS

	Command	Purpose
Step 1	**clear ssh hosts** **Example:** `switch# clear ssh hosts`	Clears the SSH host sessions.

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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **no ssh server enable**

3. **exit**
4. **show ssh server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no ssh server enable** **Example:** `switch(config)# no ssh server enable`	Disables the SSH server. The default is enabled.

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

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

Deleting SSH Server Keys

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

Note To reenable SSH, you must first generate an SSH server key (see the "Generating SSH Server Keys" section on page 5-4).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **no ssh server enable**
3. **no ssh key** [**dsa** | **rsa**]
4. **exit**
5. **show ssh key**

6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no ssh server enable** **Example:** `switch(config)# no ssh server enable`	Disables the SSH server.
Step 3	**no ssh key** [**dsa** | **rsa**] **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:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

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

Clearing SSH Sessions

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show users**
1. **clear line** *vty-line*

DETAILED STEPS

	Command	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 includes the following topics:

Enabling the Telnet Server

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **telnet server enable**
3. **exit**
4. **show telnet server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**telnet server enable** **Example:** `switch(config)# telnet server enable`	Enables the Telnet server. The default is disabled.

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

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

Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS device. You can start Telnet sessions using either IPv4 or , in Cisco NX-OS Release 4.0(2) and later releases, 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 (see the "Enabling the Telnet Server" section on page 5-11).

Enable the Telnet server on the remote device.

SUMMARY STEPS

1. **telnet** {*ipv4-address* | *hostname*} [*port-number*] [**vrf** *vrf-name*]

   telnet6 {*ipv6-address* | *hostname*} [*port-number*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command	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.
	telnet6 {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet 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. **Note** Cisco NX-OS Release 4.0(2) and later releases suport IPv6 for starting Telnet session.

Clearing Telnet Sessions

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the Telnet server on the NX-OS device.

SUMMARY STEPS

1. **show users**
2. **clear line** *vty-line*

DETAILED STEPS

	Command	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]	Displays SSH server key-pair information.
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 SSH server configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

SSH Example Configuration

To configure SSH with an OpenSSH key, follow these steps:

Step 1 Disable the SSH server.

```
switch# config t
switch(config)# no ssh server enable
```

Step 2 Generate an SSH server key.

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

Step 3 Enable the SSH server.

```
switch(config)# ssh server enable
```

Step 4 Display the SSH server key.

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

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

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

Step 5 Specify the SSH public key in OpenSSH format.

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

Step 6 Save the configuration.

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

Default Settings

Table 5-1 lists the default settings for SSH and Telnet parameters.

Table 5-1 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.

Additional References

For additional information related to implementing RBAC, see the following sections:

- Related Documents, page 5-15
- Standards, page 5-16
- MIBs, page 5-16

Related Documents

Related Topic	Document Title
Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
VRF configuration	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Standards

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

MIBs

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

CHAPTER 6

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:

Information 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.

This section includes the following topics:

About 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.

Users can have user accounts on multiple VDCs. These users can move between VDCs after an initial connection to a VDC.

Tip 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 The Cisco NX-OS software does not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally. Local users with all numeric names cannot be created. If an all numeric user name exists on an AAA server and is entered during login, the user is not logged in.

Characteristics of Strong Passwords

A strong password has the following characteristics:

- 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 (>).

Tip 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 (see the "Enabling Password-Strength Checking" section on page 6-5). Be sure to configure a strong password as shown in the sample configuration. Passwords are case sensitive.

About 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 VLANs, virtual routing and forwarding instances (VRFs), and interfaces.

The Cisco NX-OS software provides four default user roles:

- network-admin—Complete read-and-write access to the entire NX-OS device (only available in the default VDC)
- network-operator—Complete read access to the entire NX-OS device (only available in the default VDC)
- vdc-admin—Read-and-write access limited to a VDC
- vdc-operator—Read access limited to a VDC

Note You cannot change the default user roles.

You can create custom roles within a VDC. By default, the user roles that you create allow access only to the **show**, **exit**, **end**, and **configure terminal** commands. You must add rules to allow users to display or configure features.

The VDCs do not share user roles. Each VDC maintains an independent user role database. Within a VDC, roles are configured by rule and attribute assignment.

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.

About 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—Commands that apply to a function provided by the NX-OS software.
- Feature group—Default or user-defined group of features.

These 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.

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.

Virtualization Support

The users with the network-admin and network-operator roles can operate in all virtual device contexts (VDCs) when logged in from the default VDC and use the **switchto vdc** command to access other VDCs. All other user roles are local to the VDC. Roles are not shared between VDCs. Each VDC maintains an independent user role database. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Guidelines and Limitations

User accounts and RBAC have the following configuration guidelines and limitations:

- You can create up to 64 user-defined roles in a VDC in addition to the four default user roles in the default VDC and the two default user roles in the nondefault VDCs.
- You can add up to 256 rules to a user role.
- You can add up to 64 user-defined feature groups to a VDC in addition to the default feature group, L3.
- You can configure up to 256 users in a VDC.
- You can assign a maximum of 64 user roles to a user account.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

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

Note A user account must have at least one user role.

Enabling Password-Strength Checking

In Cisco NX-OS Release 4.0(3) and later releases, you can enable password-strength checking which prevents you from creating weak passwords for user accounts. For information about strong passwords, see the "Characteristics of Strong Passwords" section on page 6-2.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **password strength-check**
3. **exit**
4. **show password strength-check**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Configuring User Accounts

You can create a maximum of 256 user accounts on an NX-OS device. User accounts have the following attributes:

- Username
- Password
- Expiry date
- User roles

You can enter the password in clear text format or encrypted format. The Cisco NX-OS password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption.

User accounts can have a maximum of 64 user roles. For more information on user roles, see the "Configuring RBAC" section on page 6-8.

User accounts are local to a VDC. However, users with the network-admin or network-operator role can log in to the default VDC and access other VDCs using the **switchto vdc** command.

Note Changes to user account attributes do not take effect until the user logs in and creates a new session.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **show role**
3. **username** *user-id* [**password** [**0** | **5**]*password*] [**expire** *date*] [**role** *role-name*]
4. **exit**
5. **show user-account**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`show role`** **Example:** `switch(config)# show role`	(Optional) Displays the user roles available. You can configure other user roles, if necessary (see the "Creating User Roles and Rules" section on page 6-8)

	Command	Purpose
Step 3	**username** *user-id* [**password** [**0** | **5**] *password*] [**expire** *date*] [**role** *role-name*] **Example:** `switch(config)# username NewUser password 4Ty18Rnt`	Configure a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. **Note** Do not include the "#" or "@" character in the *user-id* argument. These characters are reserved for special use in the command-line interface (CLI). 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. For information about using SSH public keys instead of passwords, see the "Specifying the SSH Public Keys for User Accounts" section on page 5-5. 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. In the default VDC, the default role is **network-operator** if the creating user has the network-admin role, or the default role is **vdc-operator** if the creating user has the vdc-admin role. In non-default VDCs, the default user role is **vdc-operator**. **Note** The network-admin and network-operator roles are only available in the default VDC.

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

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

Configuring RBAC

This section includes the following topics:

Creating User Roles and Rules

You can configure up to 64 user roles in a VDC. Each user role can have up to 256 rules. You can assign a user role to more that 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.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **role name** *role-name*
3. **rule** *number* {**deny** | **permit**} **command** *command-string*

   rule *number* {**deny** | **permit**} {**read** | **read-write**}

   rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name*

   rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name*
4. **description** *text*
5. **exit**
6. **show role**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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.
	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.
	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.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name* **Example:** `switch(config-role)# rule 4 deny read-write 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 4	**description** *text* **Example:** `switch(config-role)# description This role does not allow users to use clear commands`	(Optional) Configures the role description. You can include spaces in the description.

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

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

Creating Feature Groups

You can create custom feature groups to add to the default list of features provided by the Cisco NX-OS software. These groups contain one or more of the features. You can create up to 64 feature groups in a VDC.

Note You cannot change the default feature group L3.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **role feature-group** *group-name*
3. **feature** *feature-name*
4. **exit**
5. **show role feature-group**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**role feature-group** *group-name*
	
	Example:
	`switch(config)# role feature 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.

	Command	Purpose
Step 3	**feature** *feature-name* **Example:** `switch(config-role-featuregrp)# feature vdc`	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	**show role feature-group** **Example:** `switch(config)# show role feature-group`	(Optional) Displays the role feature group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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 in the VDC.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

Create one or more user roles (see the "Creating User Roles and Rules" section on page 6-8).

SUMMARY STEPS

1. **config t**
2. **role name** *role-name*
3. **interface policy deny**
4. **permit interface** *interface-list*
5. **exit**
6. **show role**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show role** **Example:** `switch(config-role)# show role`	(Optional) Displays the role configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-role)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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 in the VDC.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

Create one or more user roles (see the "Creating User Roles and Rules" section on page 6-8).

SUMMARY STEPS

1. **config t**
2. **role name** *role-name*

3. **vlan policy deny**
4. **permit vlan** *vlan-range*
5. **exit**
6. **show role**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t 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:** switch(config-role-vlan)# exit switch(config-role)#	Exits role VLAN policy configuration mode.
Step 6	**show role** **Example:** switch(config-role)# show role	(Optional) Displays the role configuration.
Step 7	**copy running-config startup-config** **Example:** switch(config-role)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

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 in the VDC.

BEFORE YOU BEGIN

Ensure that you are in the desired VDC (or use the **switchto vdc** command).

Create one or more user roles (see the "Creating User Roles and Rules" section on page 6-8).

SUMMARY STEPS

1. **config t**
2. **role name** *role-name*
3. **vrf policy deny**
4. **permit vrf** *vrf-name*
5. **exit**
6. **show role**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**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	**show role** **Example:** `switch(config-role)# show role`	(Optional) Displays the role configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-role)# copy running-config startup-config`	(Optional) 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 role	Displays the user role configuration.
show role feature	Displays the feature list.
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.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example User Accounts and RBAC Configuration

The following example shows how to configure a user role:

```
role name UserA
  rule 3 permit read feature l2nac
  rule 2 permit read feature dot1x
  rule 1 deny command clear *
```

The following example shows how to configure a user role feature group:

```
role feature-group name Security-features
  feature radius
  feature tacacs
  feature dot1x
  feature aaa
  feature l2nac
  feature acl
  feature access-list
```

Default Settings

Table 6-1 lists the default settings for user accounts and RBAC parameters.

Table 6-1 Default User Accounts and RBAC Parameters

Parameters	Default
User account password	Undefined.
User account expiry date.	None.

Table 6-1 Default User Accounts and RBAC Parameters (continued)

Parameters	Default
User account role in the default VDC	Network-operator if the creating user has the network-admin role, or vdc-operator if the creating user has the vdc-admin role.
User account role in the non-VDCs	Vdc-operator if the creating user has the vdc-admin role.
Default user roles in the default VDC	Network-admin, network-operator, vdc-admin, and vdc-operator.
Default user roles in the non-default VDCs	Vdc-admin and vdc-operator.
Interface policy	All interfaces are accessible.
VLAN policy	All VLANs are accessible.
VRF policy	All VRFs are accessible.
Feature group	L3.

Additional References

For additional information related to implementing RBAC, see the following sections:

- Related Documents, page 6-16
- Standards, page 6-16
- MIBs, page 6-17

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
VRF configuration	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Standards

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

MIBs

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

CHAPTER 7

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:

Information About 802.1X

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

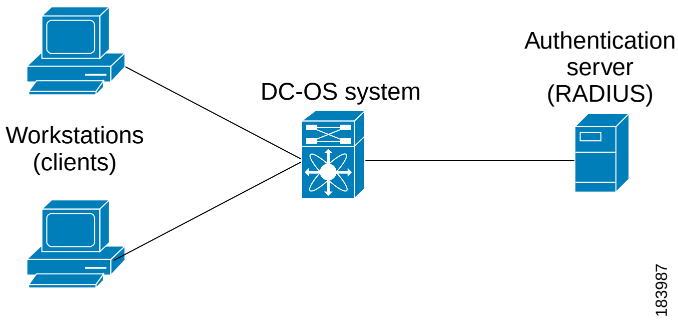
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

This section includes the following topics about 802.1X port-based authentication:

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 7-1.

Figure 7-1 802.1X Device Roles



The specific roles shown in Figure 7-1 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.

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

- Authentication server—The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the 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 Cisco Secure Access Control Server, version 3.0. RADIUS uses a supplicant-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- Authenticator—The authenticator controls the physical access to the network based on the authentication status of the supplicant. The authenticator acts as an intermediary (proxy) between the supplicant and the authentication server, requesting identity information from the supplicant, verifying the requested identity information with the authentication server, and relaying a response to the supplicant. The authenticator includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

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

Note The NX-OS device can only be a 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.

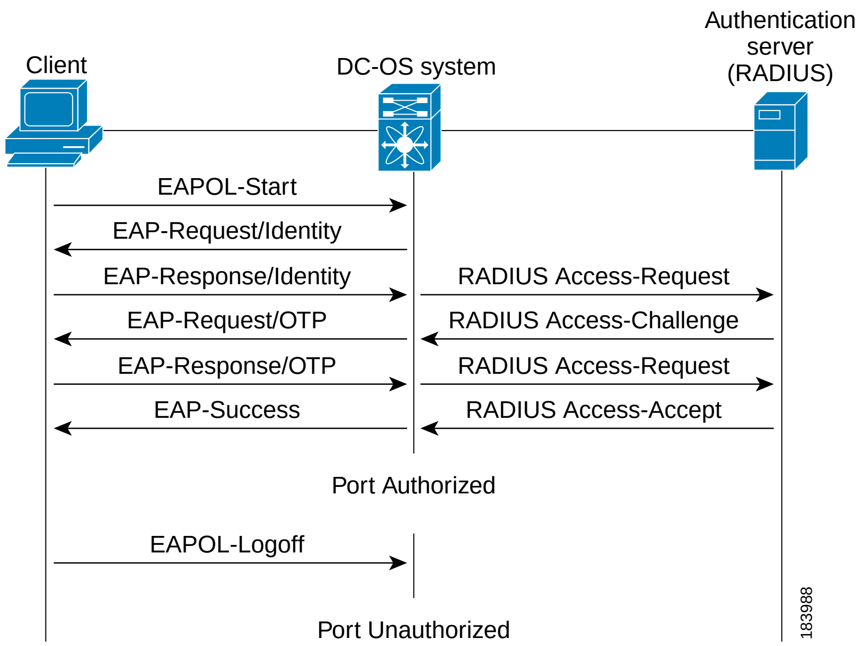
If the supplicant does not receive an EAP-request/identity frame from the authenticator during bootup, the supplicant can initiate authentication by sending an EAPOL-start frame, which prompts the authenticator to request the supplicant's identity.

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. For more information, see the "Ports in Authorized and Unauthorized States" section on page 7-4.

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. For more information, see the "Ports in Authorized and Unauthorized States" section on page 7-4.

The specific exchange of EAP frames depends on the authentication method being used. Figure 7-2 shows a message exchange initiated by the supplicant using the One-Time-Password (OTP) authentication method with a RADIUS server. 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.

Figure 7-2 Message Exchange



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 Address 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 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 authorization fails, the NX-OS device assigns the port to the guest VLAN if one is configured.

If an EAPOL packet is detected on the interface during the lifetime of the link, the 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 if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

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

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

MAC authentication bypass interacts with the features:

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

Port security—See the "802.1X with Port Security" section on page 7-6.

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.

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 shutdown upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies.

802.1X with Port Security

On NX-OS devices, you can configure 802.1X authentication and port security on the same Layer 2 ports. 802.1X uses RADIUS servers to authenticate the endpoint devices connected to a port. Port security secures ports based on MAC addresses, up to a maximum number of MAC addresses on a port. This difference allows the two features to work together. The NX-OS software supports 802.1X authentication with port security for Layer 2 ports in both host-to-switch and switch-to-switch topologies.

When 802.1X works with port security, both 802.1X and port security must authenticate supplicant MAC addresses. In multi-host mode, port security authenticates only the first supplicant MAC address. After the successful authentication of the first supplicant, the NX-OS device sends subsequent traffic from other supplicants to port security.

For more information on port security, see Chapter 13, "Configuring Port Security."

Supported Topologies

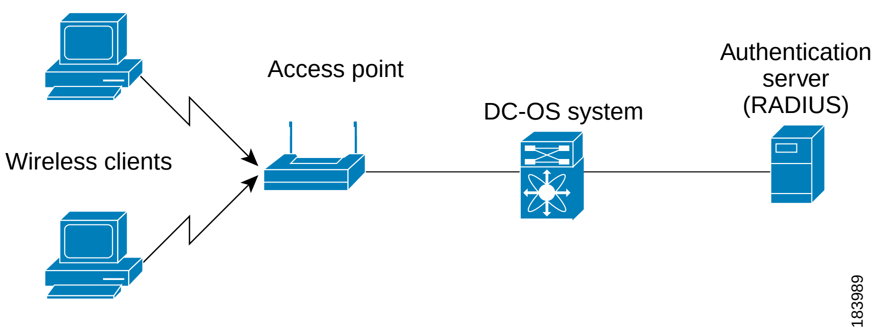
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see Figure 7-1 on page 7-2), 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.

Figure 7-3 shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one supplicant is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the NX-OS device denies access to the network to all of the attached supplicants.

Figure 7-3 Wireless LAN Example



Virtualization Support

802.1X configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for 802.1X

802.1X has the following prerequisites:

- One or more RADIUS servers accessible in the network.
- 802.1X supplicants are attached to the ports, unless you enable MAC address authentication bypass (see the "Enabling MAC Address Authentication Bypass" section on page 7-23).

802.1X Guidelines and Limitations

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

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

Configuring 802.1X

This section includes the following topics:

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

Process for Configuring 802.1X

Follow these steps to configure 802.1X authentication:

Step 1 Enable the 802.1X feature (see the "Enabling the 802.1X Feature" section on page 7-10).

Step 2 Configure the connection to the remote RADIUS server (see the "Configuring AAA Authentication Methods for 802.1X" section on page 7-11).

Step 3 Enable 802.1X authentication on the Ethernet interfaces (see the "Controlling 802.1X Authentication on an Interface" section on page 7-12).

You can perform the following optional maintenance tasks for 802.1X authentication:

- Enable periodic automatic reauthentication (see the "Enabling Global Periodic Reauthentication" section on page 7-13)
- Perform manual reauthentication (see the "Manually Reauthenticating Supplicants" section on page 7-16)
- Initialize the state of the 802.1X feature (see the "Manually Initializing 802.1X Authentication" section on page 7-17)
- Change the global 802.1X authentication timers (see the "Changing Global 802.1X Authentication Timers" section on page 7-18)
- Change the interface 802.1X authentication timers (see the "Changing 802.1X Authentication Timers for an Interface" section on page 7-19)
- Enable multiple hosts on an interface (see the "Enabling Single Host or Multiple Hosts Mode" section on page 7-22)

- Enable MAC address authentication bypass on an interface (see the "Enabling MAC Address Authentication Bypass" section on page 7-23)
- Disallow 802.1X authentication (see the "Disabling 802.1X Authentication on the NX-OS Device" section on page 7-24)
- Disable the 802.1X feature (see the "Disabling the 802.1X Feature" section on page 7-25)
- Reset the global 802.1X configuration to default values (see the "Resetting the 802.1X Global Configuration to the Default Values" section on page 7-26)
- Reset the interface 802.1X configuration to default values (see the "Resetting the 802.1X Interface Configuration to the Default Values" section on page 7-27)
- Change the frame retransmission retry count (see the "Setting the Global Maximum Authenticator-to-Supplicant Frame Retransmission Retry Count" section on page 7-28)
- Enable RADIUS accounting for 802.1X authentication (see the "Enabling RADIUS Accounting for 802.1X Authentication" section on page 7-30)
- Configure AAA accounting for 802.1X (see the "Configuring AAA Accounting Methods for 802.1X" section on page 7-31)
- Change the maximum 802.1X authentication requests (see the "Configuring the Maximum Authenticator-to-Supplicant Frame Retransmission Retry Count for an Interface" section on page 7-29)
- Change the maximum 802.1X reauthentication requests (see the "Setting the Maximum Reauthentication Retry Count on an Interface" section on page 7-32)

Enabling the 802.1X Feature

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **feature dot1x**
3. **exit**
4. **show dot1x**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

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

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

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

Step 5	`copy running-config startup-config`
	
	Example:
	`switch# copy running-config startup-config`
		(Optional) 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.

For more information on configuring RADIUS servers, see Chapter 3, "Configuring RADIUS." For information on configuring RADIUS server groups, see Chapter 2, "Configuring AAA."

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

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

SUMMARY STEPS

1. **config t**
2. **aaa authentication dot1x default group** *group-list*
3. **exit**
4. **show radius-server**
5. **show radius-server group** [*group-name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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

Step 4	**show radius-server** **Example:** `switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**show radius-server group** [*group-name*] **Example:** `switch# show radius-server group rad2`	(Optional) Displays the RADIUS server group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **dot1x port-control** {**auto** | **forced-authorized** | **forced-unauthorized**}
4. **exit**
5. **show dot1x all**
6. **show dot1x interface ethernet** *slot*/*port*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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

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

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

Enabling Global Periodic Reauthentication

You can enable global periodic 802.1X reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600 (1 hour).

To manually reauthenticate supplicants, see the "Manually Reauthenticating Supplicants" section on page 7-16.

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **dot1x re-authentication**
3. **dot1x timeout re-authperiod** *seconds*
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`dot1x re-authentication`** **Example:** `switch(config)# dot1x re-authentication`	Enables periodic reauthentication for all supplicants on the NX-OS device. By default, periodic authentication is disabled.
Step 3	**`dot1x timeout re-authperiod`** *`seconds`* **Example:** `switch(config)# dot1x timeout re-authperiod 3000`	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.

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

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

To manually reauthenticate supplicants, see the "Manually Reauthenticating Supplicants" section on page 7-16.

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **dot1x re-authentication**
4. **dot1x timeout re-authperiod** *seconds*
5. **exit**
6. **show dot1x all**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**`dot1x re-authentication`** **Example:** `switch(config-if)# dot1x re-authentication`	(Optional) Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.
Step 4	**`dot1x timeout re-authperiod`** *`seconds`* **Example:** `switch(config-if)# dot1x timeout re-authperiod 3300`	(Optional) 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)#`
		(Optional) Exits configuration mode.

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **dot1x re-authenticate** [**interface ethernet** *slot*/*port*]

DETAILED STEPS

	Command	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.

Manually Initializing 802.1X Authentication

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

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **dot1x initialize** [**interface ethernet** *slot*/*port*]

DETAILED STEPS

	Command	Purpose

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

Changing Global 802.1X Authentication Timers

The following global 802.1X authentication timers are supported on the NX-OS device:

- Quiet-period timer—When the NX-OS device cannot authenticate the supplicant, the NX-OS device 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 number smaller than the default. The default is 60 seconds. The range is from 1 to 65535.
- Switch-to-supplicant retransmission period timer—The client 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. The range is from 1 to 65535 seconds.

Note You can also configure the quiet-period timer and switch-to-supplicant transmission period timer at the interface level (see the "Changing 802.1X Authentication Timers for an Interface" section on page 7-19).

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **dot1x timeout quiet-period** *seconds*
3. **dot1x timeout tx-period** *seconds*
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**dot1x timeout quiet-period** *seconds* **Example:** `switch(config)# dot1x timeout quiet-period 30`	(Optional) Sets the number of seconds that the NX-OS device remains in the quiet state following a failed authentication exchange with any supplicant. The default is 60 seconds. The range is from 1 to 65535 seconds.
Step 3	**dot1x timeout tx-period** *seconds* **Example:** `switch(config)# dot1x timeout tx-period 20`	(Optional) Sets the number of seconds that the NX-OS device waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is 30 seconds. The range is from 1 to 65535 seconds.

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

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

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.

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **dot1x timeout quiet-period** *seconds*
4. **dot1x timeout ratelimit-period** *seconds*
5. **dot1x timeout server-timeout** *seconds*
6. **dot1x timeout supp-timeout** *seconds*
7. **dot1x timeout tx-period** *seconds*
8. **exit**
9. **show dot1x all**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 timeout quiet-period** *seconds* **Example:** `switch(config-if)# dot1x timeout quiet-period 25`	(Optional) 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	**dot1x timeout ratelimit-period** *seconds* **Example:** `switch(config-if)# dot1x timeout ratelimit-period 10`	(Optional) 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	**dot1x timeout server-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout server-timeout 60`	(Optional) Sets the number of seconds that the 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	**dot1x timeout supp-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout supp-timeout 20`	(Optional) 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	**dot1x timeout tx-period** *seconds* **Example:** `switch(config-if)# dot1x timeout tx-period 40`	(Optional) Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **dot1x host-mode** {**multi-host** | **single-host**}
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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

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

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

Enabling MAC Address Authentication Bypass

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **dot1x mac-auth-bypass** [**eap**]
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 address authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the NX-OS device to use EAP for authorization.

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

Step 5	**show dot1x all** **Example:** `switch# show dot1x all`	(Optional) Displays all 802.1X feature status and configuration information.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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 allow 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 (see the "Controlling 802.1X Authentication on an Interface" section on page 7-12). When you reenable 802.1X authentication, the NX-OS software restores the configured port mode on the interfaces.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **no dot1x system-auth-control**
3. **exit**
4. **show dot1x**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`no dot1x system-auth-control`**
	
	Example:
	`switch(config)# no dot1x`
	`system-auth-control`
		Disables 802.1X authentication on the NX-OS device. The default is enabled.
		
		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	**`show dot1x`** **Example:** `switch# show dot1x`	(Optional) Displays the 802.1X feature status.
Step 5	**`copy running-config startup-config`** **Example:** `switch# copy running-config startup-config`	(Optional) 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.

Caution Disabling 802.1X removes all 802.1X configuration from the NX-OS device. If you want to stop 802.1X authentication, see the "Disabling 802.1X Authentication on the NX-OS Device" section on page 7-24.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **no feature dot1x**
3. **exit**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

Resetting the 802.1X Global Configuration to the Default Values

You can set the 802.1X global configuration to the default values.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **dot1x default**
3. **exit**

4. **show dot1x all**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`dot1x default`** **Example:** `switch(config)# dot1x default`	Reverts to the 802.1X global configuration default values.

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the “Enabling the 802.1X Feature” section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **dot1x default**
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

Send document comments to nexus7k-docfeedback@cisco.com

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show dot1x all** **Example:** `switch(config)# show dot1x all`	(Optional) Displays all 802.1X feature status and configuration information.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Setting the Global Maximum Authenticator-to-Supplicant Frame Retransmission Retry Count

In addition to changing the authenticator-to-supplicant retransmission time, you can set the number of times that the NX-OS device sends an EAP-request/identity frame (assuming no response is received) to the supplicant before restarting the authentication process.

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **dot1x max-req** *retry-count*
3. **exit**
4. **show dot1x all**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**dot1x max-req** *retry-count* **Example:** `switch(config)# dot1x max-req 3`	Changes the maximum request retry count before restarting the 802.1X authentication process. The default is 2 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 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

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

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

You can configure 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **dot1x max-req** *count*
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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.

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **dot1x radius-accounting**
3. **exit**
4. **show dot1x**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show dot1x** **Example:** `switch# show dot1x`	(Optional) Displays the 802.1X configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **aaa accounting dot1x default group** *group-list*
3. **exit**

4. **show aaa accounting**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`aaa accounting dot1x default group`** *`group-list`* **Example:** `switch(config)# dot1x aaa accounting default group radius`	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**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of RADIUS servers for authentication.

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

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

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

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **dot1x max-reauth-req** *retry-count*
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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

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

Step 5	**show dot1x all** **Example:** `switch# show dot1x all`	(Optional) Displays all 802.1X feature status and configuration information.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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**]	Display the 802.1X feature status and configuration information for an Ethernet interface.
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying 802.1X Statistics

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **show dot1x** {**all** | **interface ethernet** *slot*/*port*} **statistics**

DETAILED STEPS

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

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

802.1X Example Configurations

The following example shows how to configure 802.1X:

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

Note Repeat the **dot1x port-control auto** command for all interfaces that require 802.1X authentication.

Default Settings

Table 7-1 lists the default settings for 802.1X parameters.

Table 7-1 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 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 NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)
Maximum retransmission number	2 times (number of times that the 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)

Additional References

For additional information related to implementing 802.1X, see the following sections:

- Related Documents, page 7-36
- Standards, page 7-36
- MIBs, page 7-36

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
VRF configuration	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Standards

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

MIBs

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

CHAPTER 8

Configuring NAC

This chapter describes how to configure Network Admission Control (NAC) on NX-OS devices.

This chapter includes the following sections:

Information About NAC

NAC allows you to check endpoint devices for security compliancy and vulnerability before these devices are allowed access to the network. This security compliancy check is referred to as *posture validation*. Posture validation allows you to prevent the spread of worms, viruses, and other rogue applications across the network.

NAC validates that the posture or state of endpoint devices complies with security policies before the devices can access protected areas of the network. For devices that comply with the security policies, NAC allows access to protected services in the network. For devices that do not comply with security policies, NAC allows access to the network only for remediation, when the posture of the device is checked again.

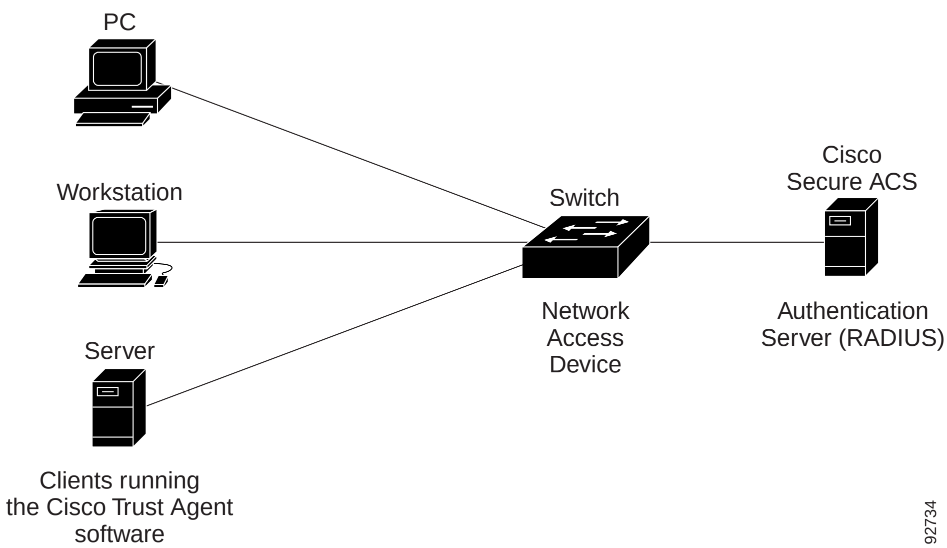
This section includes the following topics:

Send document comments to nexus7k-docfeedback@cisco.com

NAC Device Roles

NAC assigns roles to the devices in the network. Figure 8-1 shows an example of a network with the NAC device roles.

Figure 8-1 Posture Validation Devices



NAC supports the following roles for network devices:

- Endpoint device—Systems or clients on the network such as a PC, workstation, or server that is connected to an NX-OS device access port through a direct connection. The endpoint device, which is running the Cisco Trust Agent software, requests access to the LAN and switch services and responds to requests from the switch. Endpoint devices are potential sources of virus infections, and NAC must validate their antivirus statuses before granting network access.

Note The Cisco Trust Agent software is also referred to as the *posture agent* or the *antivirus client*. For more information on Cisco Trust Agent software, go to the following URL: http://www.cisco.com/en/US/products/sw/secursw/ps5057/index.html

- Network access device (NAD)— Cisco NX-OS device that provides validation services and policy enforcement at the network edge and controls the physical access to the network based on the access policy of the client. The NAD relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server.

  The NAD queries for posture credentials whenever it detects a new connection to the network. When the endpoint device has a posture agent (PA) installed, the NAD performs an in-band posture validation. The NAD acts as a relay agent between the endpoint device and AAA server for all messages in the posture validation exchange. If the NAD does not find a PA, the NAD performs an out-of-band posture validation through an audit server.

  The NAD controls which hosts have access to network destinations through that device based on a network access profile received from the AAA server once the posture validation exchange completes (whether in-band or out-of-band). The access profile can be one of the following forms:

  – VLAN or private VLAN.

- Access control list (ACL)—Determines what type of traffic for which destinations are reachable for this host in addition to any default access that is provided to all hosts independent of the NAC process (for example, access to the Dynamic Host Configuration Protocol (DHCP) server, remediation server, audit server).

The NAD triggers the posture validation process at the following times:

- When a new session starts.
- When the revalidation timer expires.
- When you enter a system administrator command.
- When the posture agent indicates that the posture has changed (only for an endpoint device with a posture agent).

For Cisco NX-OS devices, the encapsulation information in the Extensible Authentication Protocol (EAP) messages is based on the User Datagram Protocol (UDP). When using UDP, the NX-OS device uses EAP over UDP (EAPoUDP or EoU) frames.

- Authentication server— Server that performs the actual validation of the client. The authentication server validates the antivirus status of the client, determines the access policy, and notifies the NAD if the client is authorized to access the LAN and NAD services. Because the NAD acts as the proxy, the EAP message exchange between the NAD and authentication server is transparent to the NAD.

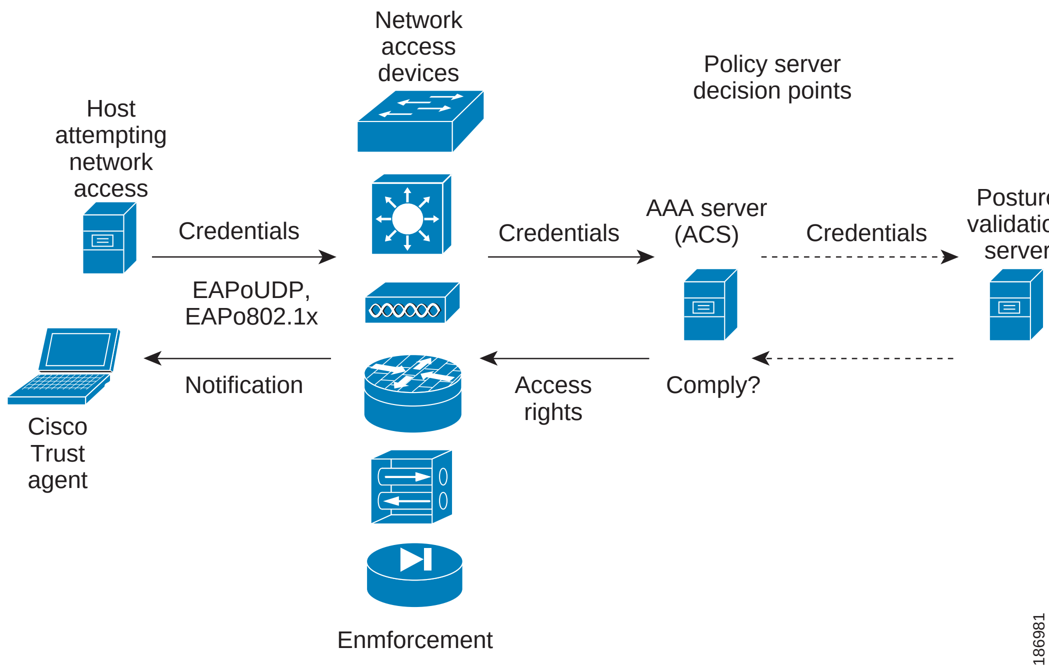
  The Cisco NX-OS device supports the Cisco Secure Access Control Server (ACS) Version 4.0 or later with RADIUS, authentication, authorization, and accounting (AAA), and EAP extensions.

- Posture validation server—Third-party server that acts as an application-specific policy decision point in NAC for authorizing a set of posture credentials against a set of policy rules. The posture validation server receives requests from an authentication server.

NAC Posture Validation

Posture validation occurs when a NAC-enabled NAD detects an endpoint device that is attempting to connect or use its network resources (see Figure 8-2). When the NAD detects a new endpoint device, it requests the network access profile for the endpoint device from an AAA server (such as the Cisco Secure ACS).

Figure 8-2 NAC Endpoint Device Posture Validation



The AAA server determines if the endpoint device has a posture agent installed. If the endpoint device has a posture agent (such as the Cisco Trust Agent), the AAA server requests the endpoint device for posture information via the NAD. The endpoint device responds to the AAA server with a set of posture credentials. The AAA server then validates the posture information locally or delegates the posture validation decisions to one or more external posture validation servers.

If the endpoint device does not have a posture agent, the AAA server may request an audit server to collect posture information from the device through other means (for example, fingerprinting and port scanning). The AAA server also asks the audit server to validate that information and return a posture validation decision.

The AAA server aggregates the posture validation results from these sources and makes an authorization decision that is based on whether the endpoint device complies with the network policy. The AAA server determines the network access profile for the endpoint device and sends the profile to the NAD for enforcement of the endpoint device authorization.

The examination of endpoint device credentials by the AAA server can result in one or more application posture tokens (APTs). An APT represents a compliance check for a given vendor's application. The AAA server aggregates all APTs from the posture validation servers into a single system posture token (SPT) that represents the overall compliance of the endpoint device. The value SPT is based on the worst APT from the set of APTs. Both APTs and SPTs are represented using the following predefined tokens:

- Healthy—The endpoint device complies with the posture policy so no restrictions are placed on this device.
- Checkup—The endpoint device is within policy but does not have the latest software; an update is recommended.
- Transition—The endpoint device is in the process of having its posture checked and is given interim access pending a result from a complete posture validation. A transition result may occur when a host is booting and complete posture information is not available, or when complete audit results are not available.

- Quarantine—The endpoint device is out of compliance and must be restricted to a quarantine network for remediation. This device is not actively placing a threat on other endpoint devices but is vulnerable to attack or infection and must be updated as soon as possible.
- Infected—The endpoint device is an active threat to other endpoint devices; network access must be severely restricted and the endpoint device must be placed into remediation or denied all network access to the endpoint device.
- Unknown—The AAA server cannot determine the posture credentials of the endpoint device. You need to determine the integrity of the endpoint device so that proper posture credentials can be attained and assessed for network access authorization.

IP Device Tracking

The IP device tracking allows endpoint devices to remain connected to the network if the AAA server is not available. Typical deployments of NAC use Cisco Secure ACS to validate the client posture and to pass policies back to the NAD.

IP device tracking provides the following benefits:

- While AAA is unavailable, the endpoint device still has connectivity to the network, although it may be restricted.
- When the AAA server is available again, a user can be revalidated and the user's policies can be downloaded from the ACS.

Note When the AAA server is down, the NAD applies the IP device tracking policy only if there is no existing policy associated with the host. Typically, during revalidation when the AAA server goes down, the NAD retains the current policies used for the endpoint device.

NAC LAN Port IP Validation

NAC LAN port IP (LPIP) validation uses the Layer 3 transport EAPoUDP to carry posture validation information. LPIP validation has the following characteristics:

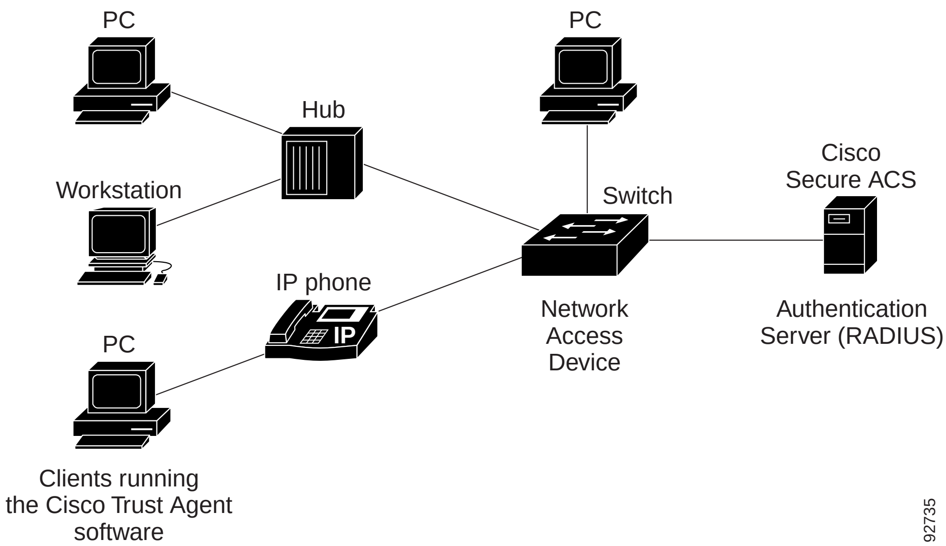
- Operates only on Layer 2 ports and cannot operate on Layer 3 ports.
- Subjects all hosts sending IP traffic on the port to posture validation.

LPIP validation triggers admission control by snooping on DHCP messages or Address Resolution Protocol (ARP) messages rather than intercepting IP packets on the data path. LPIP validation performs policy enforcement using access control lists (ACLs).

LPIP validation can process a single host connected to a NAD port or multiple hosts on the same NAD port as shown in Figure 8-3.

When you enable LPIP validation, EAPoUDP only supports IPv4 traffic. The NAD checks the antivirus status of the endpoint devices or clients and enforces access control policies.

Figure 8-3 Network Using LPIP Validation



This section describes LPIP validation and includes the following topics:

- Posture Validation, page 8-6
- Admission Triggers, page 8-6
- Posture Validation Methods, page 8-7
- Policy Enforcement Using ACLs, page 8-8
- Audit Servers and Nonresponsive Hosts, page 8-8
- NAC Timers, page 8-9
- NAC Posture Validation and Redundant Supervisor Modules, page 8-11

Posture Validation

When you enable LPIP validation on a port connected to one or more endpoint devices, the Cisco NX-OS device uses DHCP snooping and ARP snooping to identify connected hosts. The NX-OS device initiates posture validation after receiving an ARP packet or creating a DHCP snooping binding entry. ARP snooping is the default method to detect connected hosts. If you want the NAD to detect hosts when a DHCP snooping binding entry is created, you must enable DHCP snooping (see Chapter 14, "Configuring DHCP Snooping").

Admission Triggers

ARP snooping allows LPIP validation to detect hosts with either dynamically acquired or statically configured IP addresses. When the NAD receives an ARP packet from an unknown host, it triggers posture validation. If you have enabled DHCP snooping on the interface, the creation of a DHCP binding entry on the NAD triggers posture validation. DHCP snooping provides a slightly faster response time because DHCP packets are exchanged prior to sending ARP requests. Both ARP snooping and DHCP snooping can trigger posture validation on the same host. In this case, the trigger initiated by the creation of a DHCP snooping binding takes precedence over ARP snooping.

Note When you use DHCP snooping and ARP snooping to detect the presence of a host, a malicious host might set up a static ARP table to bypass posture validation. To protect against this type of exposure, you can enable IP Source Guard on the port. IP Source Guard prevents unauthorized hosts from accessing the network. (See Chapter 16, “Configuring IP Source Guard.”)

Posture Validation Methods

After posture validation is triggered for a host, you can use one of two possible methods to determine the policy to be applied for the host:

- Exception Lists, page 8-7
- EAPoUDP, page 8-7

Exception Lists

An exception list contains local profile and policy configurations. Use the identity profile to statically authorize or validate devices based on the IP address and MAC address. You can associate an identity profile with a local policy that specifies the access control attributes.

Using an exception list, you can bypass posture validation for specific endpoint devices and apply a statically configured policy. After posture validation is triggered, the NAD checks for the host information in the exception list. If a match is found in the exception list, the NAD applies the configured policy for the endpoint device.

EAPoUDP

If an endpoint device does not match the exception list, the NAD sends an EAPoUDP packet to initiate posture validation. While posture validation occurs, the NAD enforces the default access policy. After the NAD sends an EAPoUDP message to the host and the host responds to the antivirus condition request, the NAD forwards the EAPoUDP response to the Cisco Secure ACS. If the NAD does not receive a response from the host after the specified number of attempts, the NAD classifies the host as nonresponsive. After the ACS validates the credentials, the authentication server returns an Access-Accept or Access-Reject message to the NAD. The NAD updates the EAPoUDP session table and enforces the access limitations, which segments and quarantines the poorly postured endpoint device or denies network access.

Note An Access-Reject message indicates that the EAPoUDP exchange has failed. This message does not indicate that the endpoint device is poorly postured.

For an Access-Accept message, the NAD applies the enforcement policy that contains the policy-based ACL (PACL) name and starts the EAP revalidation and status query timers. For information on PACLs, see the “Policy Enforcement Using ACLs” section on page 8-8.

For an Access-Reject message, the NAD removes any enforcement policy for the host and puts the endpoint device into the Held state for a configured period of time (Hold timer). After the Hold timer expires, the NAD revalidates the endpoint device.

Note If you delete a DHCP snooping binding entry for an endpoint device, the NAD removes the client entry in the session table and the client is no longer authenticated.

Policy Enforcement Using ACLs

LPIP validation uses PACLs for policy enforcement.

The NAD applies the PACL when the posture validation fails (the AAA server sends an Access-Reject message). The default policy is to use the active MAC ACL applied to the port (also called a port ACL [PACL]). The active MAC ACL could either be a statically configured PACL or an AAA server-specified PACL based on 802.1X authentication.

The PACL defines a group that expands to a list of endpoint device IP addresses. The PACLs usually contain the endpoint device IP addresses. Once the NAD classifies an endpoint device using a particular group, the NAD adds the IP address that corresponds to the endpoint device to the appropriate group. The result is that the policy is applied to the endpoint device.

When you configure LPIP validation for an NAD port, you must also configure a default PACL on that NAD port. In addition, you should apply the default ACL to the IP traffic for hosts that have not completed posture validation.

If you configure the default ACL on the NAD and the Cisco Secure ACS sends a host access policy to the NAD, the NAD applies the policy to that traffic from the host that is connected to a NAD port. If the policy applies to the traffic, the NAD forwards the traffic. If the policy does not apply, the NAD applies the default ACL. However, if the NAD gets an endpoint device access policy from the Cisco Secure ACS but the default ACL is not configured, the LPIP validation configuration does not take effect.

Note Both DHCP snooping and ARP snooping are enabled per VLAN. However, security ACLs downloaded as a result of NAC Layer 2 posture validation are applied per port. As a result, all DHCP and ARP packets are intercepted when these features are enabled on any VLAN.

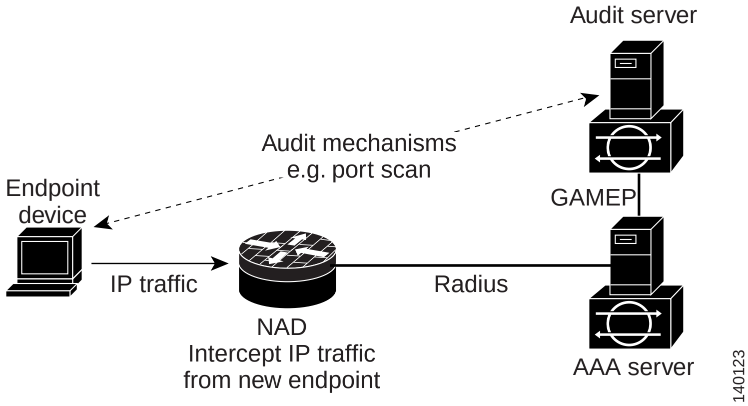
Audit Servers and Nonresponsive Hosts

Endpoint devices that do not run a posture agent (Cisco Trust Agent) cannot provide credentials when challenged by NADs. These devices are described as *agentless* or *nonresponsive*.

The NAC architecture supports audit servers to validate agentless endpoint devices. An audit server is a third-party server that can probe, scan, and determine security compliance of a host without needing a posture again on the endpoint device. The result of the audit server examination can influence the access servers to make network access policy decisions specific to the endpoint device instead of enforcing a common restrictive policy for all nonresponsive endpoint devices. You can build more robust host audit and examination functionality by integrating any third-party audit operations into the NAC architecture.

Figure 8-4 shows how audit servers fit into the typical topology.

Figure 8-4 ***NAC Device Roles***



NAC assumes that the audit server can be reached so that the endpoint device can communicate with it. When an endpoint device makes network access through the NAD configured for posture validation, the network access device eventually requests the AAA server (Cisco Secure ACS) for an access policy to be enforced for the host. The AAA server can be configured to trigger a scan of the host with an external audit server. The audit server scan occurs asynchronously and takes several seconds to complete. During the scan, the AAA server conveys a minimal restrictive security policy to NAD for enforcement along with a short poll timer (session-timeout). The NAD polls the AAA sever at the specified timer interval until the result is available from the audit server. After the AAA server receives the audit result, it computes an access policy based on the audit result and sends it to the NAD for enforcement on its next request.

NAC Timers

This section describes the NAC timers and includes the following topics:

- Hold Timer, page 8-9
- AAA Timer, page 8-10
- Retransmit Timer, page 8-10
- Revalidation Timer, page 8-10
- Status-Query Timer, page 8-10

Hold Timer

The hold timer prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Cisco Secure ACS sends an Accept-Reject message to the NAD. The default value of the hold timer is 180 seconds (3 minutes).

An EAPoUDP session might not be validated when the posture validation of the host fails, a session timer expires, or the NAD or Cisco Secure ACS receives invalid messages. If the NAD or authentication server continuously receives invalid messages, a malicious user might be trying to cause a denial-of-service attack.

AAA Timer

The AAA timer controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation. The default value of the retransmission timer is 60 seconds.

Note Setting the timer value too low might cause unnecessary transmissions; setting the timer value too high might cause poor response times.

Retransmit Timer

The retransmit timer controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation. The default value of the retransmission timer is 3 seconds.

Note Setting the timer value too low might cause unnecessary transmissions; setting the timer value too high might cause poor response times.

Revalidation Timer

The revalidation timer controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes. The timer resets when the host is revalidated. The default value of the revalidation timer is 36000 seconds (10 hours).

The Cisco NX-OS software bases the revalidation timer operation on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS-REQUEST attribute (Attribute[29]) in the Access-Accept message from the AAA server (Cisco Secure ACS). If the NAD receives the Session-Timeout value, this value overrides the revalidation timer value on the NAD.

If the revalidation timer expires, the NAD action depends on one of these values of the Termination-Action attribute:

- If the value of the Termination-Action RADIUS attribute is the default, the session ends.
- If the NAD receives a value for the Termination-Action attribute other than the default, the EAPoUDP session and the current access policy remain in effect during posture revalidation.
- If the value of the Termination-Action attribute is RADIUS, the NAD revalidates the client.
- If the packet from the server does not include the Termination-Action attribute, the EAPoUDP session ends.

Status-Query Timer

The status-query timer controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated. The default value of the status-query timer is 300 seconds (5 minutes).

The timer resets when the host is reauthenticated. When the timer expires, the NAD checks the host posture validation by sending a Status-Query message to the host. If the host sends a message to the NAD that the posture has changed, the NAD revalidates the posture of the host.

NAC Posture Validation and Redundant Supervisor Modules

When a switchover occurs, the NX-OS device maintains information about the endpoint devices and the current PACL application but loses the current state of each EAPoUDP session. The NX-OS device removes the current PACL application and restarts posture validation.

LPIP Validation and Other Security Features

This section describes how LPIP validation interacts with other security features on the NX-OS device.

This section include the following topics:

802.1X

If you configure both 802.1X and LPIP on a port, the traffic that does not pass the 802.1X-authenticated source MAC check does not trigger posture validation. When you configure 802.1X on a port, the port cannot transmit or receive traffic (other than EAP over LAN [EAPOL] frames) until the attached host is authenticated via 802.1X. This mechanism ensures that the IP traffic from the host does not trigger posture validation before it is authenticated.

Port Security

The NAD checks the source MAC against the port security MACs and drops the endpoint device if the check fails. The NAD allows posture validation only on port security-validated MAC addresses. If a port security violation occurs and results in a port shutdown, the NX-OS software removes the LPIP state of the port.

DHCP Snooping

Posture validation does not occur until after a DHCP creates a binding entry. When you enable DHCP snooping and LPIP, the NX-OS software triggers posture validation for a host when DHCP creates a binding entry for the host using DHCP to acquire IP address.

For information about DHCP snooping, see Chapter 14, “Configuring DHCP Snooping.”

Dynamic ARP Inspection

If you enable LPIP validation on the interface, posture validation is triggered only if the packet passes the dynamic ARP inspection (DAI) check. If you do not enable DAI, then all ARP packets (with valid MAC/IP pairs) will trigger posture validation.

Note ARP snooping is the default mechanism of detecting hosts. However, ARP snooping is not same as DAI. If you enable LPIP validation, the NX-OS software passes the ARP packets to LPIP validation. If you enable DAI, the NX-OS software passes the ARP packets to DAI.

Note If you have enabled DHCP snooping, the NX-OS software bypasses DAI.

For information about DAI, see Chapter 15, "Configuring Dynamic ARP Inspection."

IP Source Guard

The NX-OS software drops the packet if the source IP address is not on IP Source Guard list.

Note If you enable DHCP snooping or DAI, the NAD bypasses IP Source Guard.

Posture Host-Specific ACEs

The NX-OS software drops the packet if the packet matches the deny condition and skips the active PACL if a packet matches a permit condition. If no implicit deny exists at end of the ACEs and no match occurs, the NX-OS software checks the packet against the active PACL.

Note If you enable DHCP snooping or DAI, the NAD does not process posture host-specific ACEs.

Active PACL

The active PACL is either a statically configured PACL or an AAA server-specified PACL that is based on 802.1X authentication. Packet is dropped if matches any deny condition and moves to next step if matches a permit condition.

Note If you have enabled DHCP snooping or DAI, the NAD does not process the active PACL.

VACLs

The NX-OS software drops any packet that matches a deny condition.

Note If you have enabled DHCP snooping or DAI, the NAD bypasses the VACLs.

Virtualization Support

NAC configuration and operation are local to the virtual device context (VDC).

For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

Licensing Requirements for NAC

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	NAC requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for NAC

NAC has the following prerequisites:

- Ensure that a Layer 3 route exists between the NAD and each endpoint device.

NAC Guidelines and Limitations

NAC has the following guidelines and limitations:

- NAC uses only RADIUS for authentication.
- EAPoUDP bypass and AAA down policy are not supported.

LPIP Limitations

LPIP validation has the following limitations:

- LPIP validation is allowed only on access ports.
- You cannot enable LPIP validation on trunk ports or port channels.
- LPIP validation is not allowed on ports that are SPAN destinations.
- LPIP validation is not allowed on ports that are part of a private VLAN.
- LPIP validation does not support IPv6.
- LPIP validation is allowed only for endpoint devices directly connected to the NAD.
- You cannot use LPIP validation unless you have a Layer 3 route between the NAD and the endpoint device.

Configuring NAC

This section includes the following topics:

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

Process for Configuring NAC

Follow these steps to configure NAC:

Step 1 Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Step 2 Configure the connection to the AAA server (see the "Enabling the Default AAA Authenication Method for EAPoUDP" section on page 8-16).

Step 3 Apply PACLs to the interfaces connected to endpoint devices (see the "Applying PACLs to Interfaces" section on page 8-17).

Step 4 Enable NAC on the interfaces connected to the endpoint devices (see the "Enabling NAC on an Interface" section on page 8-19).

You can perform any of the following optional configuration tasks for NAC:

- Configure identity policies and identity profile entries for LPIP posture validation exceptions (see the "Configuring Identity Policies and Identity Profile Entries" section on page 8-20).
- Allow LPIP posture validation for clientless endpoint devices (see the "Allowing Clientless Endpoint Devices" section on page 8-22).
- Enable logging of EAPoUDP events (see the "Enabling Logging for EAPoUDP" section on page 8-23).
- Change the global maximum number of retries for EAPoUDP messages (see the "Changing the Global EAPoUDP Maximum Retry Value" section on page 8-24).
- Change the maximum number of retries for EAPoUDP messages (see the "Changing the EAPoUDP Maximum Retry Value for an Interface" section on page 8-25).
- Change the UDP port number on the NX-OS device used by EAPoUDP (see the "Changing the UDP Port for EAPoUDP" section on page 8-26).
- Configure rate limiting for EAPoUDP for simultaneous posture validation sessions (see the "Configuring Rate Limiting of Simultaneous EAPoUDP Posture Validation Sessions" section on page 8-27).
- Configure global periodic automatic LPIP posture validation for endpoint devices (see the "Configuring Global Automatic Posture Revalidation" section on page 8-28).
- Configure periodic automatic LPIP posture validation for endpoint devices on an interface (see the "Configuring Automatic Posture Revalidation for an Interface" section on page 8-29).
- Change the values of the global EAPoUDP timers used by LPIP posture validation (see the "Changing the Global EAPoUDP Timers" section on page 8-30).
- Change the values of the EAPoUDP timers for an interface used by LPIP posture validation (see the "Changing the EAPoUDP Timers for an Interface" section on page 8-32).
- Reset the EAPoUDP global configuration to the default values (see the "Resetting the EAPoUDP Global Configuration to the Default Values" section on page 8-34).
- Reset the EAPoUDP configuration on an interface to the default values (see the "Resetting the EAPoUDP Interface Configuration to the Default Values" section on page 8-35).
- Configure IP device tracking (see the "Configuring IP Device Tracking" section on page 8-36).
- Manually initialize some or all EAPoUDP sessions (see the "Manually Initializing EAPoUDP Sessions" section on page 8-39).
- Manually revalidate some or all EAPoUDP sessions (see the "Manually Revalidating EAPoUDP Sessions" section on page 8-40).

Enabling EAPoUDP

The NX-OS device relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server. You must enable EAPoUDP before configuring NAC on the NX-OS device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **feature eou**
3. **exit**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**feature eou** **Example:** `switch(config)# feature eou`	Enables EAPoUDP. The default is disabled.

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

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

Enabling the Default AAA Authenication Method for EAPoUDP

You must enable the default AAA authentication method EAPoUDP. For more information on AAA authentication methods, see Chapter 2, "Configuring AAA." For information on configuring RADIUS servers, see Chapter 3, "Configuring RADIUS."

Note LPIP can use only RADIUS for authentication.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **config t**
2. **aaa authentication eou default group** *group-list*
3. **exit**
4. **show aaa authentication**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**aaa authentication eou default group** *group-list* **Example:** `switch(config)# aaa authentication eou default group RadServer`	Configures a list of one or more RADIUS server groups as the default AAA authentication method for EAPoUDP. The *group-list* argument consists of a space-delimited list of group. The group names are as follows: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of RADIUS servers for authentication. The default setting is no method.

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

Step 4	**show aaa authentication** **Example:** `switch# show aaa authentication`	(Optional) Displays the default AAA authentication methods.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying PACLs to Interfaces

You must apply a PACL to the access interfaces on the NAD that perform LPIP posture validation if no PACL is available from the AAA server.

For more information on PACLs, see Chapter 11, "Configuring MAC ACLs."

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Create a MAC ACL.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **mac access-group** *access-list*
4. **exit**
5. **show running-config interface**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**`mac access-group`** *`access-list`* **Example:** `switch(config-if)# mac access-group acl-01`	Applies a PACL to the interface for traffic that flows in the direction specified. **Note** An interface can have only one PACL. To replace the PACL on the interface, enter this command again using the new PACL name.

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

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

Enabling NAC on an Interface

You must enable NAC on an interface for posture validation to occur.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **switchport**
4. **switchport mode access**
5. **nac enable**
6. **exit**
7. **show running-config interface**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Sets the interface as a Layer 2 switching interface. By default, all ports are Layer 3 ports.
Step 4	**switchport mode access** **Example:** `switch(config-if)# switchport mode access`	Configures the port mode as access.
Step 5	**nac enable** **Example:** `switch(config-if)# nac enable`	Enables NAC on the interface.

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

	Command	Purpose
Step 7	**show running-config interface** **Example:** `switch(config)# show running-config interface`	(Optional) Displays the interface PACL configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Identity Policies and Identity Profile Entries

You can use the identity profile to configure exceptions to LPIP posture validation. The identity profile contains entries for the endpoint devices for which are not subject to LPIP validation. You can optionally configure an identity policy for each identity profile entry that specifies a PACL that the NX-OS device applies to the endpoint device. The default identity policy is the PACL for the interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **identity policy** *policy-name*
3. **object-group** *access-list*
4. **description** "*text*"
5. **exit**
6. **show identity policy**
7. **identity profile eapoudp**
8. **device** {**authenticate** | **not-authenticate**} {**ip-address** *ipv4-address* [*ipv4-subnet-mask*] | **mac-address** *mac-address* [*mac-subnet-mask*]} **policy** *name*
9. **exit**
10. **show identity profile eapoudp**
11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**identity policy** *policy-name*
	
	Example:
	`switch(config)# identity policy AccType1`
	`switch(config-id-policy)#`
		Specifies the identity policy name and enters identity policy configuration mode. You can create a maximum of 1024 identity policies. The maximum length of the name is 100 characters.

Step 3	**object-group** *access-list* **Example:** `switch(config-id-policy)# object-group maxaclx`	Specifies the IP ACL or MAC ACL for the policy.
Step 4	**description "***text***"** **Example:** `switch(config-id-policy)# description "This policy prevents endpoint device without a PA"`	(Optional) Provides a description for the identity policy. The maximum length is 100 characters.

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

Step 6	**show identity policy** **Example:** `switch(config)# show identity policy`	(Optional) Displays the identity policy configuration.

Step 7	**identity profile eapoudp**
	
	Example:
	`switch(config)# identity profile eapoudp`
	`switch(config-id-prof)#`
		Enters identity profile configuration mode for EAPoUDP.

Step 8	**device** {**authenticate** | **not-authenticate**} {**ip-address** *ipv4-address* [*ipv4-subnet-mask*] | **mac-address** *mac-address* [*mac-subnet-mask*]} **policy** *name* **Example:** `switch(config-id-prof)# device authenticate ip-address 10.10.2.2 policy AccType1`	Specifies an exception entry. The maximum number of entries is 5000.

Step 9	**exit**
	
	Example:
	`switch(config-id-prof)# exit`
	`switch(config)#`
		Exits identity profile configuration mode.

	Command	Purpose
Step 10	**show identity profile eapoudp** **Example:** `switch(config)# show identity profile eapoudp`	(Optional) Displays the identity profile configuration.
Step 11	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Allowing Clientless Endpoint Devices

You can allow posture validation endpoint devices in your network that do not have a posture agent installed (clientless). The posture validation is performed by an audit server that has access to the endpoint devices.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the “Enabling EAPoUDP” section on page 8-15).

Verify that the AAA server and clientless endpoint devices can access the audit server.

SUMMARY STEPS

1. **config t**
2. **eou allow clientless**
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou allow clientless** **Example:** `switch(config)# eou allow clientless`	Allows posture validation for clientless endpoint devices. The default is disabled.

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

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

Enabling Logging for EAPoUDP

You can enable logging for EAPoUDP event messages. EAPoUDP events include errors and status changes. The destination for these event messages is the configured syslog.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou logging**
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou logging** **Example:** `switch(config)# eou logging`	Enables EAPoUDP logging. The default is disabled.

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

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

Changing the Global EAPoUDP Maximum Retry Value

You can change the global maximum number of EAPoUDP retries. The default value is three.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou max-retry** *count*
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou max-retry** *count* **Example:** `switch(config)# eou max-retry 2`	Changes the EAPoUDP maximum retry count. The default is 3. The range is from 1 to 3.

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

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

Changing the EAPoUDP Maximum Retry Value for an Interface

You can change the maximum number of EAPoUDP retries for an interface. The default value is three.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Enable NAC on the interface (see the "Enabling NAC on an Interface" section on page 8-19)

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **eou max-retry** *count*
4. **exit**
5. **show eou**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**eou max-retry** *count* **Example:** `switch(config-if)# eou max-retry 2`	Changes the EAPoUDP maximum retry count. The default is 3. The range is from 1 to 3.

	Command	Purpose

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

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

Changing the UDP Port for EAPoUDP

You can change the UDP port used by EAPoUDP. The default port is 21862.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the “Enabling EAPoUDP” section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou port** *udp-port*
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou port** *udp-port* **Example:** `switch(config)# eou port 27180`	Changes the UDP port used by EAPoUDP. The default is 21862. The range is from 1 to 65535.

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

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

Configuring Rate Limiting of Simultaneous EAPoUDP Posture Validation Sessions

You can configure rate limiting to control the number of simultaneous EAPoUDP posture validations sessions. You can change the rate-limiting value that controls the maximum number of simultaneous EAPoUDP posture validation sessions. The default number is 20. Setting the number to zero (0) disables rate limiting.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou ratelimit** *number-of-sessions*
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou ratelimit** *number-of-sessions* **Example:** `switch(config)# eou ratelimit 15`	Configures the number of simultaneous EAPoUDP posture validation sessions. The default is 20. The range is from 0 to 200. **Note** A setting of zero (0) disables rate limiting.

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

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

Configuring Global Automatic Posture Revalidation

The NX-OS software automatically revalidates the posture of the endpoint devices for the NX-OS device at a configured interval. The default interval is 36,000 seconds (10 hours). You can disable revalidation or change the length of the revalidation interval.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou revalidate**
3. **eou timeout revalidation** *seconds*
4. **exit**
5. **show eou**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou revalidate** **Example:** `switch(config)# eou revalidate`	(Optional) Enables the automatic posture validation. The default is enabled.
Step 3	**eou timeout revalidation** *seconds* **Example:** `switch(config)# eou timeout revalidation 30000`	(Optional) Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds. Use the **no eou revalidate** command to disable automatic posture validation.

	Command	Purpose
Step 4	**exit** **Example:** switch(config)# exit switch#	Exits global configuration mode.
Step 5	**show eou** **Example:** switch# show eou	(Optional) Displays the EAPoUDP configuration.
Step 6	**copy running-config startup-config** **Example:** switch# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Automatic Posture Revalidation for an Interface

The NX-OS software automatically revalidates the posture of the endpoint devices for the NX-OS device at a configured interval. The default interval is 36,000 seconds (10 hours). You can disable revalidation or change the length of the revalidation interval.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Enable NAC on the interface (see the "Enabling NAC on an Interface" section on page 8-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **eou revalidate**
4. **eou timeout revalidation** *seconds*
5. **exit**
6. **show eou**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**eou revalidate** **Example:** `switch(config-if)# eou revalidate`	(Optional) Enables the automatic posture validation. The default is enabled. Use the **no eou revalidate** command to disable automatic posture validation.
Step 4	**eou timeout revalidation** *seconds* **Example:** `switch(config-if)# eou timeout revalidation 30000`	(Optional) Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.

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

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

Changing the Global EAPoUDP Timers

The NX-OS software supports the following global timers for EAPoUDP:

- AAA—Controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation.
- Hold period—Prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Cisco Secure ACS sends an Accept-Reject message to the NAD.
- Retransmit—Controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation.

- Revalidation—Controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes.
- Status query—Controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated.

For more information on these timers, see the "NAC Timers" section on page 8-9.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou timeout aaa** *seconds*
3. **eou timeout hold-period** *seconds*
4. **eou timeout retransmit** *seconds*
5. **eou timeout revalidation** *seconds*
6. **eou timeout status-query** *seconds*
7. **exit**
8. **show eou**
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	**eou timeout aaa** *seconds* **Example:** switch(config)# eou timeout aaa 30	(Optional) Changes the AAA timeout interval. The default is 60 seconds (1 minute). The range is from 0 to 60 seconds.
Step 3	**eou timeout hold-period** *seconds* **Example:** switch(config)# eou timeout hold-period 300	(Optional) Changes the hold period timeout interval. The default is 180 seconds (3 minutes). The range is from 60 to 86400 seconds.
Step 4	**eou timeout retransmit** *seconds* **Example:** switch(config)# eou timeout retransmit 10	(Optional) Changes the retransmit timeout interval. The default is 3 seconds. The range is from 1 to 60 seconds.

	Command	Purpose
Step 5	**eou timeout revalidation** *seconds* **Example:** `switch(config)# eou timeout revalidation 30000`	(Optional) Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.
Step 6	**eou timeout status-query** *seconds* **Example:** `switch(config)# eou timeout status-query 360`	(Optional) Changes the status query timeout interval. The default is 300 seconds (5 minutes). The range is from 10 to 1800 seconds.

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

Step 8	**show eou** **Example:** `switch# show eou`	(Optional) Displays the EAPoUDP configuration.
Step 9	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing the EAPoUDP Timers for an Interface

The NX-OS software supports the following timers for EAPoUDP for each interface enabled for NAC:

- AAA—Controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation.
- Hold period—Prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Cisco Secure ACS sends an Accept-Reject message to the NAD.
- Retransmit—Controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation.
- Revalidation—Controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes.
- Status query—Controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated.

For more information on these timers, see the "NAC Timers" section on page 8-9.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Enable NAC on the interface (see the "Enabling NAC on an Interface" section on page 8-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **eou timeout aaa** *seconds*
4. **eou timeout hold-period** *seconds*
5. **eou timeout retransmit** *seconds*
6. **eou timeout revalidation** *seconds*
7. **eou timeout status-query** *seconds*
8. **exit**
9. **show eou**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**eou timeout aaa** *seconds* **Example:** `switch(config-if)# eou timeout aaa 50`	(Optional) Changes the AAA timeout interval. The default is 60 seconds (1 minute). The range is from 0 to 60 seconds.
Step 4	**eou timeout hold-period** *seconds* **Example:** `switch(config-if)# eou timeout hold-period 300`	(Optional) Changes the hold period timeout interval. The default is 180 seconds (3 minutes). The range is from 60 to 86400 seconds.
Step 5	**eou timeout retransmit** *seconds* **Example:** `switch(config-if)# eou timeout retransmit 10`	(Optional) Changes the retransmit timeout interval. The default is 3 seconds. The range is from 1 to 60 seconds.
Step 6	**eou timeout revalidation** *seconds* **Example:** `switch(config-if)# eou timeout revalidation 30000`	(Optional) Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.
Step 7	**eou timeout status-query** *seconds* **Example:** `switch(config-if)# eou timeout status-query 360`	(Optional) Changes the status query timeout interval. The default is 300 seconds (5 minutes). The range is from 10 to 1800 seconds.

	Command	Purpose

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

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

Resetting the EAPoUDP Global Configuration to the Default Values

You can reset the EAPoUDP global configuration to the default values.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **config t**
2. **eou default**
3. **exit**
4. **show eou**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**eou default** **Example:** `switch(config)# eou default`	Resets the EAPoUDP configuration to the default values.

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

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

Resetting the EAPoUDP Interface Configuration to the Default Values

You can reset the EAPoUDP configuration for an interface to the default values.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

Enabled NAC on the interface (see the "Enabling NAC on an Interface" section on page 8-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **eou default**
4. **exit**
5. **show eou**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**eou default** **Example:** `switch(config-if)# eou default`	Resets the EAPoUDP configuration for the interface to the default values.

	Command	Purpose

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

Step 5	**show eou interface ethernet** *slot*/*port* **Example:** `switch(config)# show eou interface ethernet 2/1`	(Optional) Displays the EAPoUDP configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring IP Device Tracking

You can configure IP device tracking. The process for the IP device tracking for AAA servers operates is as follows:

1. The NX-OS device detects a new session.
2. Before posture validation is triggered and if the AAA server is unreachable, the NX-OS device applies the IP device tracking policy and maintains the session state as AAA DOWN.
3. When the AAA server is once again available, a revalidation occurs for the host.

Note When the AAA server is down, the NX-OS device applies the IP device tracking policy only if no existing policy is associated with the endpoint device. During revalidation when the AAA server goes down, the NX-OS device retains the policies that are used for the endpoint device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **ip device tracking enable**
3. **ip device tracking probe** {**count** *count* | **interval** *seconds*}
4. **radius server host** {*hostname* | *ip-address*} **text** [**username** *username* [**password** *password*]] [**idle-time** *minutes*]
5. **exit**
6. **show ip device tracking all**
7. **show radius-server** {*hostname* | *ip-address*}
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**ip device tracking enable** **Example:** `switch(config)# ip device tracking enable`	Enables the IP device tracking. The default state is enabled.
Step 3	**ip device tracking probe** {**count** *count* | **interval** *seconds*} **Example:** `switch(config)# ip device tracking probe count 4`	(Optional) Configures these parameters for the IP device tracking table: • **count** *count*—Sets the number of times that the NX-OS device sends the ARP probe. The range is from 1 to 5. The default is 3. • **interval** *interval*—Sets the number of seconds that the NX-OS device waits for a response before resending the ARP probe. The range is from 1 to 302300 seconds. The default is 30 seconds.
Step 4	**radius-server host** {*hostname* | *ip-address*} **test** [**username** *username* [**password** *password*]] [**idle-time** *minutes*] **Example:** `switch(config)# radius-server host 10.10.1.1 test username User2 password G1r2D37&k idle-time 5`	(Optional) Configures RADIUS server test packet parameters. The default username is test and the default password is test. The **idle-time** parameter determines how often the server is tested to determine its operational status. If there is no traffic to the RADIUS server, the NAD sends dummy packets to the RADIUS server based on the idle timer value. The default value for the idle timer is 0 minutes (disabled). If you have multiple RADIUS servers, reenter this command.

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

Step 6	**show ip device tracking all** **Example:** `switch# show ip device tracking all`	(Optional) Displays IP device tracking information.
Step 7	**show radius-server** {*hostname* | *ip-address*} **Example:** `switch# show radius-server 10.10.1.1`	(Optional) Displays RADIUS server information.
Step 8	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Clearing IP Device Tracking

You can clear IP device tracking information for AAA servers.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **clear ip device tracking all**
2. **clear ip device tracking interface ethernet** *slot/port*
3. **clear ip device tracking ip-address** *ipv4-address*
4. **clear ip device tracking mac-address** *mac-address*
5. **show ip device tracking all**

DETAILED STEPS

	Command	Purpose

Step 1	`clear ip device tracking all`
	
	Example:
	`switch# clear ip device tracking all`
		(Optional) Clears all EAPoUDP sessions.

Step 2	`clear ip device tracking interface ethernet` *slot/port* **Example:** `switch# clear ip device tracking interface ethernet 2/1`	(Optional) Clears EAPoUDP sessions on a specified interface.
Step 3	`clear ip device tracking ip-address` *ipv4-address* **Example:** `switch# clear ip device tracking ip-address 10.10.1.1`	(Optional) Clears an EAPoUDP session for a specified IPv4 address in the format A.B.C.D.
Step 4	`clear ip device tracking mac-address` *mac-address* **Example:** `switch# clear ip device tracking mac-address 000c.30da.86f4`	(Optional) Clears an EAPoUDP session for a specified MAC address in the format XXXX.XXXX.XXXX.

Step 5	`show ip device tracking all`
	
	Example:
	`switch# show ip device tracking all`
		(Optional) Displays IP device tracking information.

Send document comments to nexus7k-docfeedback@cisco.com

Manually Initializing EAPoUDP Sessions

You can manually initialize EAPoUDP sessions.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **eou initialize all**
2. **eou initialize authentication** {**clientless** | **eap** | **static**}
3. **eou initialize interface ethernet** *slot*/*port*
4. **eou initialize ip-address** *ipv4-address*
5. **eou initialize mac-address** *mac-address*
6. **eou initialize posturetoken** *name*
7. **show eou all**

DETAILED STEPS

	Command	Purpose
Step 1	**eou initialize all** **Example:** `switch# eou initialize all`	(Optional) Initializes all EAPoUDP sessions.
Step 2	**eou initialize authentication** {**clientless** | **eap** | **static**} **Example:** `switch# eou initialize authentication static`	(Optional) Initializes EAPoUDP sessions with a specified authentication type.
Step 3	**eou initialize interface ethernet** *slot*/*port* **Example:** `switch# eou initialize interface ethernet 2/1`	(Optional) Initializes EAPoUDP sessions on a specified interface.
Step 4	**eou initialize ip-address** *ipv4-address* **Example:** `switch# eou initialize ip-address 10.10.1.1`	(Optional) Initializes an EAPoUDP session for a specified IPv4 address in the format A.B.C.D.
Step 5	**eou initialize mac-address** *mac-address* **Example:** `switch# eou initialize mac-address 000c.30da.86f4`	(Optional) Initializes an EAPoUDP session for a specified MAC address in the format XXXX.XXXX.XXXX.

	Command	Purpose
Step 6	**eou initialize posturetoken** *name* **Example:** `switch# eou initialize posturetoken Healthy`	(Optional) Initializes an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	**show eou all** **Example:** `switch# show eou all`	(Optional) Displays the EAPoUDP session configuration.

Manually Revalidating EAPoUDP Sessions

You can manually revalidate EAPoUDP sessions.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **eou revalidate all**
2. **eou revalidate authentication** {**clientless** | **eap** | **static**}
3. **eou revalidate interface ethernet** *slot/port*
4. **eou revalidate ip-address** *ipv4-address*
5. **eou revalidate mac-address** *mac-address*
6. **eou revalidate posturetoken** *name*
7. **show eou all**

DETAILED STEPS

	Command	Purpose
Step 1	**eou revalidate all** **Example:** `switch# eou revalidate all`	(Optional) Revalidates all EAPoUDP sessions.
Step 2	**eou revalidate authentication** {**clientless** | **eap** | **static**} **Example:** `switch# eou revalidate authentication static`	(Optional) Revalidates EAPoUDP sessions with a specified authentication type.
Step 3	**eou revalidate interface ethernet** *slot/port* **Example:** `switch# eou revalidate interface ethernet 2/1`	(Optional) Revalidates EAPoUDP sessions on a specified interface.

	Command	Purpose
Step 4	**eou revalidate ip-address** *ipv4-address* **Example:** `switch# eou revalidate ip-address 10.10.1.1`	(Optional) Revalidates an EAPoUDP session for a specified IPv4 address.
Step 5	**eou revalidate mac-address** *mac-address* **Example:** `switch# eou revalidate mac-address 000c.30da.86f4`	(Optional) Revalidates an EAPoUDP session for a specified MAC address.
Step 6	**eou revalidate posturetoken** *name* **Example:** `switch# eou revalidate posturetoken Healthy`	(Optional) Revalidates an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	**show eou all** **Example:** `switch# show eou all`	(Optional) Displays the EAPoUDP session configuration.

Clearing EAPoUDP Sessions

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable EAPoUDP (see the "Enabling EAPoUDP" section on page 8-15).

SUMMARY STEPS

1. **clear eou all**
2. **clear eou authentication** {**clientless** | **eap** | **static**}
3. **clear eou interface ethernet** *slot/port*
4. **clear eou ip-address** *ipv4-address*
5. **clear eou mac-address** *mac-address*
6. **clear eou posturetoken** *name*
7. **show eou all**

DETAILED STEPS

	Command	Purpose
Step 1	**clear eou all** **Example:** `switch# clear eou all`	(Optional) Clears all EAPoUDP sessions.
Step 2	**clear eou authentication** {**clientless** | **eap** | **static**} **Example:** `switch# clear eou authentication static`	(Optional) Clears EAPoUDP sessions with a specified authentication type.
Step 3	**clear eou interface ethernet** *slot*/*port* **Example:** `switch# clear eou interface ethernet 2/1`	(Optional) Clears EAPoUDP sessions on a specified interface.
Step 4	**clear eou ip-address** *ipv4-address* **Example:** `switch# clear eou ip-address 10.10.1.1`	(Optional) Clears an EAPoUDP session for a specified IPv4 address.
Step 5	**clear eou mac-address** *mac-address* **Example:** `switch# clear eou mac-address 000c.30da.86f4`	(Optional) Clears an EAPoUDP session for a specified MAC address.
Step 6	**clear eou posturetoken** *name* **Example:** `switch# clear eou posturetoken Healthy`	(Optional) Clears an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	**show eou all** **Example:** `switch# show eou all`	(Optional) Displays the EAPoUDP session configuration.

Disabling the EAPoUDP Feature

You can disable the EAPoUDP feature on the NX-OS device.

Caution Disabling EAPoUDP removes all EAPoUDP configuration from the NX-OS device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Enable the 802.1X feature on the NX-OS device (see the "Enabling the 802.1X Feature" section on page 7-10).

SUMMARY STEPS

1. **config t**
2. **no feature eou**

3. **exit**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no feature eou** **Example:** `switch(config)# no feature eou`	Disables EAPoUDP. **Caution** Disabling the EAPoUDP feature removes all EAPoUDP configuration.

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

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

Verifying the NAC Configuration

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

Command	Purpose
show eou [**all** | **authentication** {**clientless** | **eap** | **static**} | **interface ethernet** *slot/port* | **ip-address** *ipv4-address* | **mac-address** *mac-address* | **posturetoken** *name*]	Displays the EAPoUDP configuration.
show ip device tracking [**all** | **interface ethernet** *slot/port* | **ip-address** *ipv4-address* | **mac-address** *mac-address*]	Displays IP device tracking information.
show running-config eou [**all**]	Displays the EAPoUDP configuration in the running configuration.
show startup-config eou	Displays the EAPoUDP configuration in the startup configuration.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example NAC Configuration

The following example shows how to configure NAC:

```
feature eou
aaa authentication eou default group radius
mac access-list macacl-01
  10 permit any any 0x100
interface Ethernet8/1
  mac access-group macacl-01
```

Default Settings

Table 8-1 lists the default settings for NAC parameters.

Table 8-1 Default NAC Parameters

Parameters	Default
EAPoUDP	Disabled.
EAP UDP port number	21862 (0x5566).
Clientless hosts allowed	Disabled.
Automatic periodic revalidation	Enabled.
Revalidation timeout interval	36000 seconds (10 hours).
Retransmit timeout interval	3 seconds.
Status query timeout interval	300 seconds (5 minutes).
Hold timeout interval	180 seconds (3 minutes).
AAA timeout interval	60 seconds (1 minute).
Maximum retries	3.
EAPoUDP rate limit maximum	20 simultaneous sessions.
EAPoUDP logging	Disabled.
IP device tracking	Enabled.

Additional References

For additional information related to implementing NAC, see the following sections:

- Related Documents, page 8-45

Related Documents

Related Topic	Document Title
Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

Send document comments to nexus7k-docfeedback@cisco.com

CHAPTER 9

Configuring Cisco TrustSec

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

This chapter includes the following sections:

Information About Cisco TrustSec

This section includes the following topics:

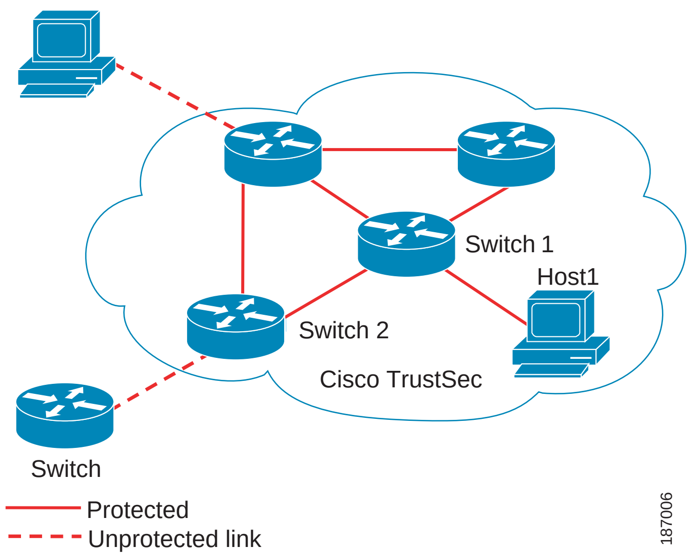
Cisco TrustSec Architecture

The Cisco TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Each device in the cloud is authenticated by its neighbors. Communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and data-path replay protection mechanisms. Cisco TrustSec also uses the device and user identification information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Cisco TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

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

Figure 9-1 shows an example of a Cisco TrustSec cloud. In this example, several networking devices and an endpoint device are inside the Cisco TrustSec cloud. One endpoint device and one networking device are outside the cloud because they are not Cisco TrustSec-capable devices or they have been refused access.

Figure 9-1 Cisco TrustSec Network Cloud Example



The Cisco TrustSec architecture consists of the following major components:

- Authentication—Verifies the identity of each device before allowing them to join the Cisco TrustSec network.
- Authorization—Decides the level of access to the Cisco TrustSec network resources for a device based on the authenticated identity of the device.
- Access Control—Applies access policies on per-packet basis using the source tags on each packet.
- Secure communication—Provides encryption, integrity, and data-path replay protection for the packets that flow over each link in the Cisco TrustSec network.

A Cisco TrustSec network has the following three entities:

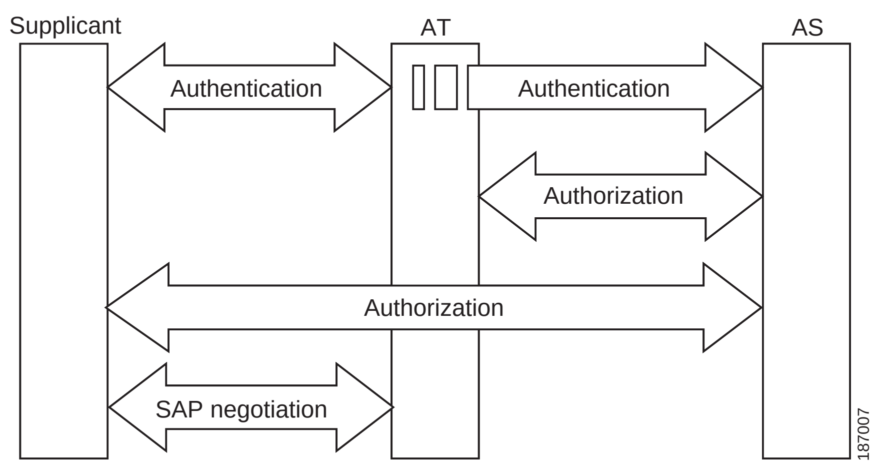
- Supplicants—Devices that attempt to join a Cisco TrustSec network.
- Authenticators (AT)—Devices that are already part of a Cisco TrustSec network.
- Authorization server—Servers that may provide authentication information, authorization information, or both.

When the link between the supplicant and the AT first comes up, the following sequence of events may occur:

1. Authentication (802.1X)—The authentication server performs the authentication of the supplicant or the authentication completes trivially if you configure the devices to unconditionally authenticate each other.
2. Authorization—Each side of the link obtains policies, such as SGT and ACLs, that to apply to the link. A supplicant may need to use the AT as a relay if it has no other Layer 3 route to the authentication server.
3. Security Association Protocol (SAP) negotiation—The EAPOL-Key exchange occurs between the supplicant and the AT to negotiate a cipher suite, exchange security parameter indexes (SPIs), and manage keys. Successful completion of all three tasks results in the establishment of a security association (SA).

Ports stay in unauthorized state (blocking state) until the SAP negotiation completes (see Figure 9-2).

Figure 9-2 SAP Negotiation



SAP negotiation can use any of the following modes of operation:

- Galois/Counter Mode (GCM) encryption
- GCM authentication (GMAC)
- No encapsulation (clear text)
- Encapsulation with no encryption or authentication

Based on the IEEE 802.1AE standard, Cisco TrustSec uses ESP-128 GCM and GMAC.

Authentication

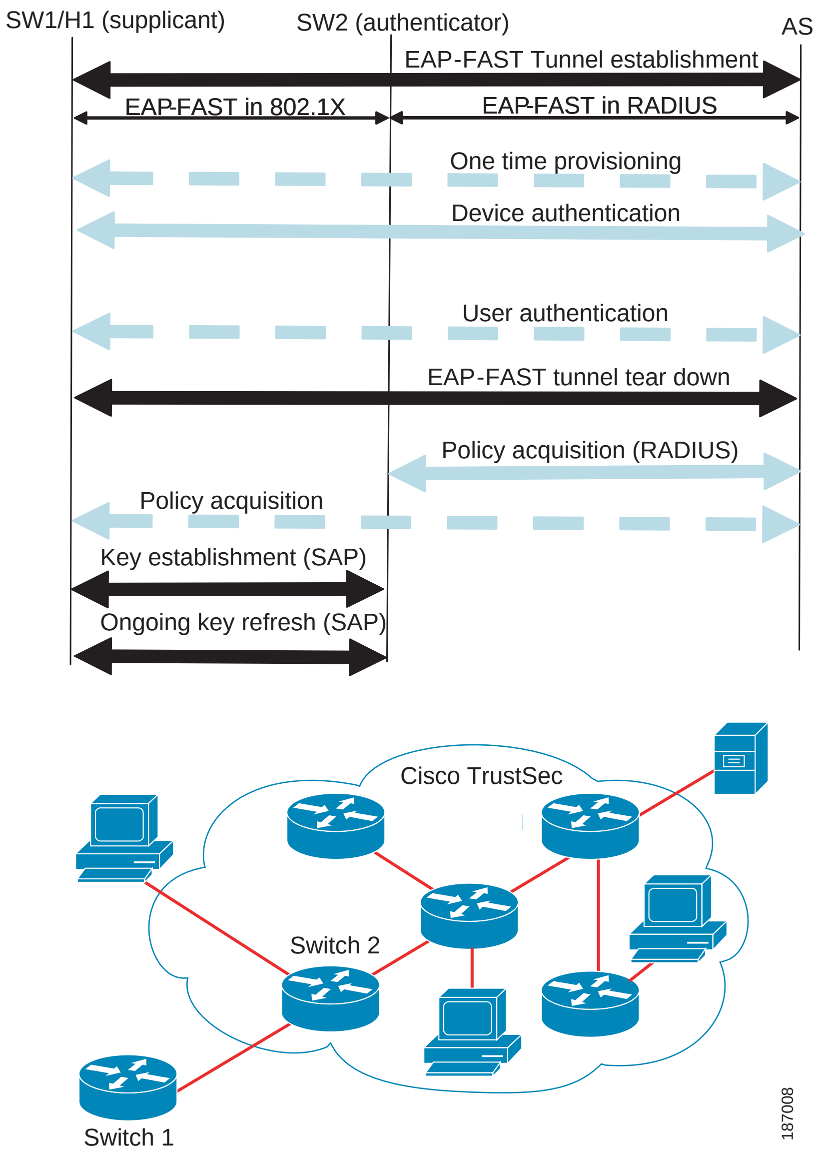
Cisco TrustSec authenticates a device before allowing it to join the network. Cisco TrustSec uses 802.1X authentication with Extensible Authentication Protocol Flexible Authentication via Secure Tunnel (EAP-FAST) as the Extensible Authentication Protocol (EAP) method to perform the authentication.

This section includes the following topics:

Cisco TrustSec and Authentication

Cisco TrustSec uses EAP-FAST for authentication. EAP-FAST conversations allow for other EAP method exchanges inside the EAP-FAST tunnel using chains. This allows administrators to use traditional user authentication methods, such as Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAPv2), while still having security provided by the EAP-FAST tunnel. Figure 9-3 shows the EAP-FAST tunnel and inner methods as used in Cisco TrustSec.

Figure 9-3 Cisco TrustSec Authentication



This section includes the following topics:

- Cisco TrustSec Enhancements to EAP-FAST, page 9-5
- 802.1x Role Selection, page 9-5
- Cisco TrustSec Authentication Summary, page 9-5

Cisco TrustSec Enhancements to EAP-FAST

The implementation of EAP-FAST for Cisco TrustSec has the following enhancements:

- Authenticate the authenticator—Securely determines the identity of the AT by requiring the AT to use its protected access credential (PAC) to derive the shared secret between itself and the authentication server. This feature also prevents you from configuring RADIUS shared secrets on the authentication server for every possible IP address that can be used by the AT.
- Notify each peer of the identity of its neighbor—By the end of the authentication exchange, the authentication server has identified both the supplicant and the AT. The authentication server conveys the identity of the AT, and whether the AT is Cisco TrustSec-capable, to the supplicant by using additional type-length-value parameters (TLVs) in the protected EAP-FAST termination. The authentication server also conveys the identity of the supplicant and whether the supplicant is Cisco TrustSec-capable, to the AT by using RADIUS attributes in the Access- Accept message. Because each peer knows the identity of its neighbor, it can send additional RADIUS Access-Requests to the authentication server to acquire the policy to be applied on the link.
- AT posture evaluation—The AT provides its posture information to the authentication server whenever it starts the authentication exchange with the authentication server on behalf of the supplicant.

802.1x Role Selection

In 802.1X, the AT must have IP connectivity with the authentication server because it has to relay the authentication exchange between the supplicant and the AT using RADIUS over UDP/IP. When an endpoint device, such as a PC, connects to a network, it is obvious that it should act as a supplicant. However, in the case of a Cisco TrustSec connection between two network devices, the 802.1X role of each network device might not be immediately apparent to the other network device.

Instead of requiring manual configuration of the AT and supplicant roles for the NX-OS devices, Cisco TrustSec runs a role-selection algorithm to automatically determine which NX-OS device acts as the AT and which acts as the supplicant. The role-selection algorithm assigns the AT role to the device that has IP reachability to a RADIUS server. Both devices start both the AT and supplicant state machines. When an NX-OS device detects that its peer has access to a RADIUS server, it terminates its own AT state machine and assumes the role of the supplicant. If both NX-OS devices have access to a RADIUS server, the algorithm compares the MAC addresses used as the source for sending the EAP over LAN (EAPOL) packets. The NX-OS device that has the MAC address with the higher value becomes the AT and the other NX-OS device becomes the supplicant.

Cisco TrustSec Authentication Summary

By the end of the Cisco TrustSec authentication process, the authentication server has performed the following actions:

- Verified the identities of the supplicant and the AT.
- Authenticated the user if the supplicant is an endpoint device.

At the end of the Cisco TrustSec authentication process, both the AT and the supplicant know following:

- Device ID of the peer
- Cisco TrustSec capability information of the peer
- Key used for the SAP

Device Identities

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

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

Device Credentials

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

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

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

User Credentials

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

SGACLs and SGTs

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

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

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

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

Cisco TrustSec defines access control policies between the security groups. By assigning devices within the network to security groups and applying access control between and within the security groups, Cisco TrustSec essentially achieves access control within the network. Figure 9-4 shows an example of an SGACL policy.

Figure 9-4 SGACL Policy Example

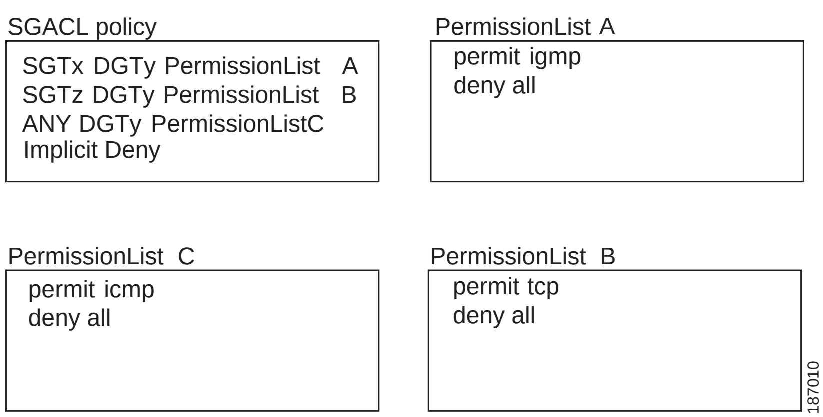
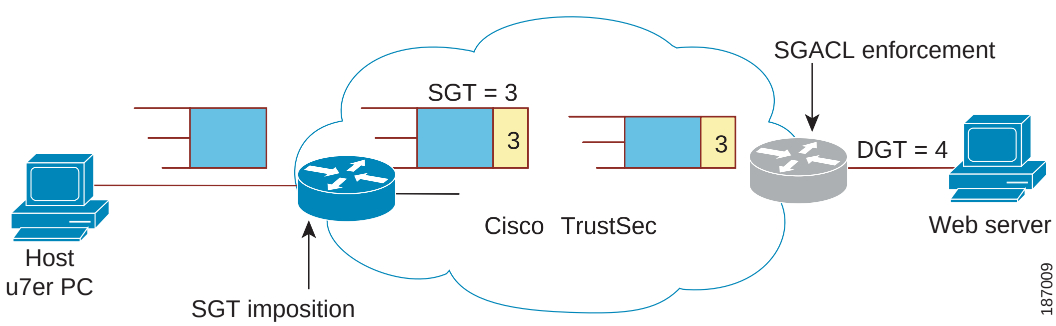


Figure 9-5 shows how the SGT assignment and the SGACL enforcement operate in a Cisco TrustSec network.

Figure 9-5 SGT and SGACL in Cisco TrustSec Network



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

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

# of ACEs = (# of sources specified) X (# of destinations specified) X (# of permissions specified)

In Cisco TrustSec uses the following formula:

# of ACEs = # of permissions specified

This section includes the following topics:

- Determining the Source Security Group, page 9-8
- Determining the Destination Security Group, page 9-8
- SXP for SGT Propagation Across Legacy Access Networks, page 9-9

Determining the Source Security Group

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

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

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

Determining the Destination Security Group

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

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

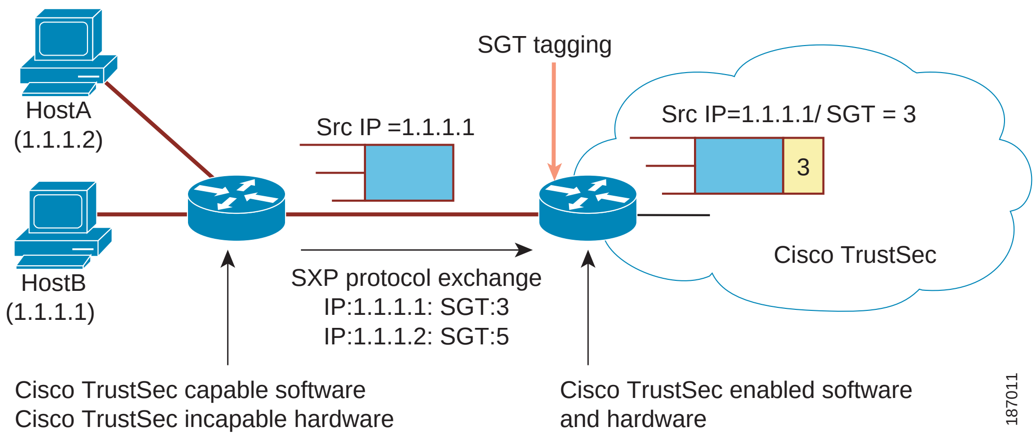
- Destination SGT of the egress port obtained during policy acquisition
- Destination SGT lookup based on the destination IP address

SXP for SGT Propagation Across Legacy Access Networks

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

SXP operates between access layer devices and distribution layer devices. The access layer devices use SXP to pass the IP addresses of the Cisco TrustSec authenticated devices along with their SGTs to the distribution switches. Distribution devices with both Cisco TrustSec-enable software and hardware can use this information to tag packets appropriately and enforce SGACL policies (see Figure 9-6).

Figure 9-6 SXP Protocol to Propagate SGT information



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

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

Authorization and Policy Acquisition

After authentication ends, both the supplicant and AT obtain the security policy from the authentication server. The supplicant and AT enforce the policy against each other. Both the supplicant and AT provide the peer device ID that each receives after authentication. If the peer device ID is not available, Cisco TrustSec can use a manually configured peer device ID.

The authentication server returns the following policy attributes:

- Cisco TrustSec trust—Indicates whether the neighbor device is to be trusted for the purpose of putting the SGT in the packets.

- Peer SGT—Indicates the security group that the peer belongs to. If the peer is not trusted, all packets received from the peer are tagged with this SGT. If the device does not know if the SGACLs are associated with the peer's SGT, the device may send a follow-up request to fetch the SGACLs.
- Authorization expiry time—Indicates the number of seconds before the policy expires. The Cisco-proprietary attribute-value (AV) pairs indicates the expiration time of an authorization or policy response to a Cisco TrustSec device. A Cisco TrustSec device should refresh its policy and authorization before it times out.

Tip Each Cisco TrustSec device should support some minimal default access policy in case it is not able to contact the authentication server to get an appropriate policy for the peer.

Environment Data Download

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

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

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

- Server lists—List of servers that the client can use for future RADIUS requests (for both authentication and authorization).
- Device SGT—Security group to which the device itself belongs.
- Expiry timeout—Interval that controls how often the Cisco TrustSec device should refresh its environment data.

RADIUS Relay Functionality

The NX-OS device that plays the role of the Cisco TrustSec AT in the 802.1X authentication process has IP connectivity to the authentication server, which allows it to acquire the policy and authorization from the authentication server by exchanging RADIUS messages over UDP/IP. The supplicant device may not have IP connectivity with the authentication server. In such cases, Cisco TrustSec allows the AT to act as a RADIUS relay for the supplicant.

The supplicant sends a special EAP over LAN (EAPOL) message to the Cisco TrustSec AT that contains the RADIUS server IP address and UDP port and the complete RADIUS request. The Cisco TrustSec AT extracts the RADIUS request from the received EAPOL message and sends it over UDP/IP to the authentication server. When the RADIUS response returns from the authentication server, the Cisco TrustSec AT forwards the message back to the supplicant, encapsulated in an EAPOL frame.

Virtualization Support

Cisco TrustSec configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

Licensing Requirements for Cisco TrustSec

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Creating nondefault VDCs requires an Advanced Services license. For a complete explanation of the NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*. **Note** Cisco TrustSec licensing does not have a grace period. You must obtain and install an Advanced Services license before you can use Cisco TrustSec.

Prerequisites for Cisco TrustSec

Cisco TrustSec has the following prerequisites:

- You must install the Advance Service license.
- You must enable the 802.1X feature.

Guidelines and Limitations

Cisco TrustSec has the following guidelines and limitations:

- Cisco TrustSec uses RADIUS for authentication.
- You cannot configure both Cisco TrustSec and 802.1X on an interface; you can configure only one or the other. However, you must enable the 802.1X feature for Cisco TrustSec to use EAP-FAST authentication.
- AAA authentication and authorization for Cisco TrustSec is only supported by the Cisco Secure Access Control Server (ACS).
- Cisco TrustSec supports IPv4 addressing only.
- SXP cannot use the management (mgmt 0) interface.
- You cannot enable Cisco TrustSec on interfaces in half-duplex mode.

Configuring Cisco TrustSec

This section includes the following topics:

Enabling the Cisco TrustSec Feature

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

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **feature dot1x**
3. **feature cts**
4. **exit**
5. **show cts**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

	Command	Purpose
Step 3	**feature cts** **Example:** `switch(config)# feature cts`	Enables the Cisco TrustSec feature.

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

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

Configuring Cisco TrustSec Device Credentials

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

Note You must also configure the Cisco TrustSec credentials for the NX-OS device on the Cisco Secure ACS (see the *Configuration Guide for the Cisco Secure ACS*).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **config t**
2. **cts device-id** *name* **password** *password*
3. **exit**
4. **show cts**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**cts device-id** *name* **password** *password* **Example:** `switch(config)# cts device-id MyDevice1 password Cisco321`	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive.

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

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

Configuring AAA for Cisco TrustSec

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

Note Only the Cisco Secure ACS supports Cisco TrustSec.

For more information on configuring RADIUS servers, see Chapter 3, "Configuring RADIUS." For information on configuring RADIUS server groups, see Chapter 2, "Configuring AAA."

This section includes the following sections:

- Configuring AAA on the Cisco TrustSec Seed NX-OS Device, page 9-15
- Configuring AAA on Cisco TrustSec Nonseed NX-OS Devices, page 9-17

Configuring AAA on the Cisco TrustSec Seed NX-OS Device

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

Note When you configure the AAA RADIUS server group for the seed NX-OS device, you must specify a VRF. If you use the management VRF, no further configuration is necessary for the nonseed devices in the network cloud. If you use a different VRF, you must configure the nonseed devices with that VRF (see the Configuring AAA on Cisco TrustSec Nonseed NX-OS Devices, page 9-17).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Obtain the IPv4 or IPv6 address or hostname for the Cisco ACS.

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **password** *password* **pac**
3. **show radius-server**
4. **aaa group server radius** *group-name*
5. **server** {*ipv4-address* | *ipv6-address* | *hostname*}
6. **use-vrf** *vrf-name*
7. **exit**
8. **aaa authentication dot1x default group** *group-name*
9. **aaa authorization cts default group** *group-name*
10. **exit**
11. **show radius-server groups** [*group-name*]
12. **show aaa authentication**
13. **show aaa authorization**
14. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **password** *password* **pac** **Example:** `switch(config)# radius-server host 10.10.1.1 password L1a0K2s9 pac`	Configures a RADIUS server host with a password and PAC.
Step 3	**show radius-server** **Example:** `switch# show radius-server`	(Optional) Displays the RADIUS server configuration.

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

Step 5	**server** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** `switch(config-radius)# server 10.10.1.1`	Specifies the RADIUS server host address.
Step 6	**use-vrf** *vrf-name* **Example:** `switch(config-radius)# use-vrf management`	Specifies the management VRF for the AAA server group. **Note** If you use the management VRF, no further configuration is necessary for the nonseed devices in the network cloud. If you use a different VRF, you must configure the nonseed devices with that VRF (see the Configuring AAA on Cisco TrustSec Nonseed NX-OS Devices, page 9-17).

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

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

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

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

Step 11	**show radius-server groups** [*group-name*] **Example:** `switch# show radius-server group rad2`	(Optional) Displays the RADIUS server group configuration.
Step 12	**show aaa authentication** **Example:** `switch# show aaa authentication`	(Optional) Displays the AAA authentication configuration.
Step 13	**show aaa authorization** **Example:** `switch# show aaa authorization`	(Optional) Displays the AAA authorization configuration.
Step 14	**show cts pacs** **Example:** `switch# show show cts pacs`	(Optional) Displays the Cisco TrustSec PAC information.
Step 15	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring AAA on Cisco TrustSec Nonseed NX-OS Devices

Cisco TrustSec configures an AAA server group named aaa-private-sg on the nonseed NX-OS devices in the network cloud. By default, the aaa-private-sg server group uses the management VRF to communicate with the Cisco Secure ACS and no further configuration is required on the nonseed NX-OS devices. However, if you choose to use a different VRF, you must change the aaa-private-sg on the nonseed NX-OS device to use the correct VRF.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you have configured a seed NX-OS device in your network (see Configuring AAA on the Cisco TrustSec Seed NX-OS Device, page 9-15).

SUMMARY STEPS

1. **config t**
2. **aaa group server radius aaa-private-sg**

3. **use-vrf** *vrf-name*
4. **exit**
5. **show radius-server groups** [*group-name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`aaa group server radius aaa-private-sg`**
	
	Example:
	`switch(config)# aaa group server radius aaa-private-sg`
	`switch(config-radius)#`
		Specifies the RADIUS server group aaa-private-sg and enters RADIUS server group configuration mode.

Step 3	**`use-vrf`** *`vrf-name`* **Example:** `switch(config-radius)# use-vrf MyVRF`	Specifies the management VRF for the AAA server group.

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

Step 5	**`show radius-server groups aaa-private-sg`** **Example:** `switch(config)# show radius-server groups aaa-private-sg`	(Optional) Displays the RADIUS server group configuration for the default server group.
Step 6	**`copy running-config startup-config`** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Cisco TrustSec Authentication, Authorization, SAP, and Data Path Security

This section includes the following topics:

- Enabling Cisco TrustSec Authentication, page 9-19
- Configuring Data-Path Replay Protection for Cisco TrustSec on Interfaces, page 9-21
- Configuring SAP Operation Modes for Cisco TrustSec on Interfaces, page 9-23
- Configuring SGT Propagation for Cisco TrustSec on Interfaces, page 9-25
- Regenerating SAP Keys on an Interface, page 9-26

Cisco TrustSec Configuration Process for Cisco TrustSec Authentication and Authorization

Follow these steps to configure Cisco TrustSec authentication and authorization:

Step 1 Enable the Cisco TrustSec feature (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Step 2 Enable Cisco TrustSec authentication (see the "Enabling Cisco TrustSec Authentication" section on page 9-19).

Step 3 Enable 802.1X authentication for Cisco TrustSec on the interfaces (see the "Enabling Cisco TrustSec Authentication" section on page 9-19).

Enabling Cisco TrustSec Authentication

You must enable Cisco TrustSec authentication on the interfaces. By default, the data path replay protection feature is enabled and the SAP operating mode is GCM-encrypt.

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

Note Enabling 802.1X mode for Cisco TrustSec automatically enables authorization and SAP on the interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **no data-path replay protection**
5. **sap modelist** {**gmc-encrypt** | **gmac** | **no-encap** | **null**}
6. **exit**
7. **shutdown**
8. **no shutdown**
9. **exit**
10. **show cts interface all**
11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface ethernet** *slot*/*port* [- *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Cisco TrustSec and enters Cisco TrustSec 802.1X configuration mode.

Step 4	**no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	(Optional) Disables replay protection. The default is enabled.
Step 5	**sap modelist** {**gcm-encrypt** | **gmac** | **no-encap** | **null**} **Example:** `switch(config-if-cts-dot1x)# sap modelist gcm-encrypt`	(Optional) Configures the SAP operation mode on the interface. • **gcm-encrypt**—GCM encryption • **gmac**—GCM authentication only • **no-encap**— No encapsulation for SAP and no SGT insertion • **null**— Encapsulation without authentication or encryption The default is **gcm-encrypt**.

Step 6	**exit**
	
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Cisco TrustSec 802.1X configuration mode.

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

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

	Command	Purpose
Step 10	**show cts interface all** **Example:** `switch(config)# show cts interface all`	(Optional) Displays the Cisco TrustSec configuration on the interfaces.
Step 11	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Data-Path Replay Protection for Cisco TrustSec on Interfaces

By default, the NX-OS software enables the data-path reply protection feature. You can disable the data-path replay protection feature on the interfaces for Layer 2 Cisco TrustSec if the connecting device does not support SAP.

Caution For the data-path replay protection configuration to take affect, you must enable and disable the interface which disrupts traffic on the interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec authentication on the interface (see the "Enabling Cisco TrustSec Authentication" section on page 9-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port* [- *port2*]
3. **cts dot1x**
4. **no replay-protection**
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. **show cts interface all**
10. **copy running-config startup-config**

Send document comments to nexus7k-docfeedback@cisco.com

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface ethernet** *slot*/*port* [- *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Cisco TrustSec and enters Cisco TrustSec 802.1X configuration mode.

Step 4	**no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	Disables data-path replay protection. The default is enabled. Use the **replay-protection** command to enable data-path replay protection on the interface.

Step 5	**exit**
	
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Cisco TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and disables the data-path reply protection feature on the interface.

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

Step 9	**show cts interface all** **Example:** `switch(config)# show cts interface all`	(Optional) Displays the Cisco TrustSec configuration on the interface.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring SAP Operation Modes for Cisco TrustSec on Interfaces

You can configure the SAP operation mode on the interfaces for Layer 2 Cisco TrustSec. The default SAP operation mode is GCM-encrypt.

Caution For the SAP operation mode configuration to take affect, you must enable and disable the interface which disrupts traffic on the interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec authentication on the interface (see the "Enabling Cisco TrustSec Authentication" section on page 9-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port* [- *port2*]
3. **cts dot1x**
4. **sap modelist gcm-encrypt**
   sap modelist gmac
   sap modelist no-encap
   sap modelist null
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. **show cts interface all**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface ethernet** *slot*/*port* [- *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single interface or a range of interfaces and enters interface configuration mode.

	Command	Purpose

Step 3	**cts dot1x**
	
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Cisco TrustSec and enters Cisco TrustSec 802.1X configuration mode.

Step 4	**sap modelist gcm-encrypt** **Example:** `switch(config-if-cts-dot1x)# sap modelist gcm-encrypt`	Configures GCM encryption mode for SAP on the interface. The default is **gcm-encrypt**.
	sap modelist gmac **Example:** `switch(config-if-cts-dot1x)# sap modelist gmac`	Configures GCM authentication only mode for SAP on the interface.
	sap modelist no-encap **Example:** `switch(config-if-cts-dot1x)# sap modelist no-encap`	Configures no encapsulation for SAP on the interface and does not insert an SGT.
	sap modelist null **Example:** `switch(config-if-cts-dot1x)# sap modelist null`	Configures encapsulation without authentication or encryption for SAP on the interface. Only the SGT is encapsulated.

Step 5	**exit**
	
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Cisco TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and SAP operation mode on the interface.

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

Step 9	**show cts interface all** **Example:** `switch(config)# show cts interface all`	(Optional) Displays the Cisco TrustSec configuration on the interface.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring SGT Propagation for Cisco TrustSec on Interfaces

SGT propagation feature on the Layer 2 interface is enabled by default. You can disable the SGT propagation feature on an interface if the peer device connected to the interface can not handle Cisco TrustSec packets tagged with an SGT.

Caution For the SGT propagation configuration to take affect, you must enable and disable the interface which disrupts traffic on the interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec authentication on the interface (see the "Enabling Cisco TrustSec Authentication" section on page 9-19).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port* [- *port2*]
3. **cts dot1x**
4. **no propagate-sgt**
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. **show cts interface all**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface ethernet** *slot*/*port* [- *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Cisco TrustSec and enters Cisco TrustSec 802.1X configuration mode.

	Command	Purpose
Step 4	**no propagate-sgt** **Example:** `switch(config-if-cts-dot1x)# no propagate-sgt`	Disables SGT propagation. The default is enabled. Use the **propagate-sgt** command to enable SGT propagation on the interface.

Step 5	**exit**
	
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Cisco TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and disables the data-path reply protection feature on the interface.

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

Step 9	**show cts interface all** **Example:** `switch(config)# show cts interface all`	(Optional) Displays the Cisco TrustSec configuration on the interface.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Regenerating SAP Keys on an Interface

You can trigger an SAP protocol exchange to generate a new set of keys and protect the data traffic flowing on an interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **cts rekey ethernet** *slot/port*
2. **show cts interface all**

DETAILED STEPS

	Command	Purpose
Step 1	**cts rekey ethernet** *slot/port* **Example:** `switch# cts rekey ethernet 2/3`	Generates the SAP keys for an interface.
Step 1	**show cts interface all** **Example:** `switch# show cts interface all`	(Optional) Displays Cisco TrustSec configuration on the interfaces.

Configuring Cisco TrustSec Authentication in Manual Mode

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

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

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **cts manual**
4. **sap pmk** {*key* | **use-dot1x**} [**modelist** {**gcm-encrypt** | **gmac** | **no-encap** | **null**}]
5. **policy dynamic identity** *peer-name*

   policy static sgt *tag* [**trusted**]
6. **exit**
7. **shutdown**
8. **no shutdown**
9. **exit**
10. **show cts interface all**

11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

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

Step 3	**`cts manual`**
	
	Example:
	`switch(config-if)# cts manual`
	`switch(config-if-cts-manual)#`
		Enters Cisco TrustSec manual configuration mode.
		
		Note You cannot enable Cisco TrustSec on interfaces in half-duplex mode.

Step 4	**`sap pmk`** {*`key`* | **`use-dot1x`**} [**`modelist`** {**`gcm-encrypt`** | **`gmac`** | **`no-encap`** | **`null`**}] **Example:** `switch(config-if-cts-manual)# sap pmk fedbaa modelist gmac`	Configures the SAP pairwise master key (PMK) and operation mode. SAP is disabled by default in Cisco TrustSec manual mode. The *key* argument is a hexadecimal value with an even number of characters and a maximum length of 32 characters. Use the **use-dot1x** keyword when the peer device does not support Cisco TrustSec 802.1X authentication or authorization but does support SAP data path encryption and authentication. The mode list configures the cipher mode for the data path encryption and authentication as follows: • **gcm-encrypt**—GCM encryption mode • **gmac**—GCM authentication mode • **no-encap**—No encapsulation and no SGT insertion • **null**— Encapsulation of the SGT without authentication or encryption The default mode is **gcm-encrypt**.

	Command	Purpose
Step 5	**policy dynamic identity** *peer-name* **Example:** `switch(config-if-cts-manual)# policy dynamic identity MyDevice2`	Configures dynamic authorization policy download. The *peer-name* argument is the Cisco TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Cisco TrustSec credentials (see "Configuring Cisco TrustSec Device Credentials" section on page 9-13) and AAA for Cisco TrustSec (see "Configuring AAA for Cisco TrustSec" section on page 9-14).
	policy static sgt *tag* [**trusted**] **Example:** `switch(config-if-cts-manual)# policy static sgt 0x03`	Configures a static authorization policy. The *tag* argument is in hexadecimal format and the range is from 0x0 to 0xffff. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden.

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

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

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

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

Configuring SGACL Policies

This section includes the following topics:

- SGACL Policy Configuration Process, page 9-30
- Enabling SGACL Policy Enforcement on VLANs, page 9-30

SGACL Policy Configuration Process

Follow these steps to configure Cisco TrustSec SGACL policies:

Step 1 For Layer 2 interfaces, enable SGACL policy enforcement for the VLANs with Cisco TrustSec-enabled interfaces (see the "Enabling SGACL Policy Enforcement on VLANs" section on page 9-30).

Step 2 For Layer 3 interfaces, enable SGACL policy enforcement for the VRFs with Cisco TrustSec-enabled interfaces (see the "Enabling SGACL Policy Enforcement on VRFs" section on page 9-31).

Step 3 If you are not using AAA on a Cisco Secure ACS to download the SGACL policy configuration, manually configure the SGACL mapping and policies (see the "Manually Configuring IPv4-Address-to-SGACL SGT Mapping" section on page 9-33 and the "Manually Configuring SGACL Policies" section on page 9-35).

Enabling SGACL Policy Enforcement on VLANs

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **config t**
2. **vlan** *vlan-id*
3. **cts role-based enforcement**
4. **exit**
5. **show cts role-based enable**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

Step 3	**cts role-based enforcement** **Example:** `switch(config-vlan)# cts role-based enforcement`	Enables Cisco TrustSec SGACL policy enforcement on the VLAN.

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

Step 5	**show cts role-based enable** **Example:** `switch(config)# show cts role-based enable`	(Optional) Displays the Cisco TrustSec SGACL enforcement configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling SGACL Policy Enforcement on VRFs

If you use SGACLs, you must enable SGACL policy enforcement in the VRFs that have Cisco TrustSec-enabled Layer 3 interfaces.

Note You cannot enable SGACL policy enforcement on the management VRF.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled dynamic Address Resolution Protocol (ARP) inspection (see Chapter 15, "Configuring Dynamic ARP Inspection") or Dynamic Host Configuration Protocol (DHCP) snooping (see Chapter 14, "Configuring DHCP Snooping").

SUMMARY STEPS

1. **config t**
2. **vrf context** *vrf-name*
3. **cts role-based enforcement**
4. **exit**
5. **show cts role-based enable**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

Step 3	**cts role-based enforcement** **Example:** `switch(config-vrf)# cts role-based enforcement`	Enables Cisco TrustSec SGACL policy enforcement on the VRF.

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

Step 5	**show cts role-based enable** **Example:** `switch(config)# show cts role-based enable`	(Optional) Displays the Cisco TrustSec SGACL enforcement configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Manually Configuring Cisco TrustSec SGTs

You can manually configure unique Cisco TrustSec security group tags (SGTs) for the packets subject to SGACL enforcement.

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **cts sgt** *tag*
3. **exit**
4. **show cts environment-data**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	`cts sgt` *`tag`* **Example:** `switch(config)# cts device-id MyDevice1 password Cisco321`	Configures the SGT for packets sent from the device. The *tag* argument is a hexadecimal value in the format **0x***hhhh*. The range is from 0x1 to 0xfffd.

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

Step 4	`show cts environment-data`
	
	Example:
	`switch# show cts environment-data`
		(Optional) Displays the Cisco TrustSec environment data information.

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

Manually Configuring IPv4-Address-to-SGACL SGT Mapping

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SGACL policy enforcement on the VLAN (see the "Enabling SGACL Policy Enforcement on VLANs" section on page 9-30) or VRF (see the "Enabling SGACL Policy Enforcement on VRFs" section on page 9-31).

SUMMARY STEPS

1. **config t**
2. **vlan** *vlan-id*

   vrf context *vrf-name*
3. **cts role-based sgt-map** *ipv4-address tag*
4. **exit**
5. **show cts role-based enable**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

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

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

	Command	Purpose

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

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

Step 5	**show cts role-based sgt-map** **Example:** `switch(config)# show cts role-based sgt-map`	(Optional) Displays the Cisco TrustSec SGACL SGT mapping configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Manually Configuring SGACL Policies

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SGACL policy enforcement on the VLAN (see the "Enabling SGACL Policy Enforcement on VLANs" section on page 9-30) and VRF (see the "Enabling SGACL Policy Enforcement on VRFs" section on page 9-31).

SUMMARY STEPS

1. **config t**
2. **cts role-based access-list** *list-name*
3. **deny all**

   deny icmp

   deny igmp

   deny ip

   deny tcp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]

   deny udp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]
4. **permit all**

   permit icmp

   permit igmp

permit ip

permit tcp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]

permit udp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]

5. **exit**
6. **cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name*
7. **show cts role-based access-list**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

Step 3	**deny all** **Example:** `switch(config-rbacl)# deny all`	Denies all traffic.
	deny icmp **Example:** `switch(config-rbacl)# deny icmp`	Denies Internet Control Message Protocol (ICMP) traffic.
	deny igmp **Example:** `switch(config-rbacl)# deny igmp`	Denies Internet Group Management Protocol (IGMP) traffic.
	deny all **Example:** `switch(config-rbacl)# deny ip`	Denies IP traffic.
	deny tcp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# deny tcp src lt 10`	Denies TCP traffic. The default denies all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
	deny udp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# deny udp dest eq 100`	Permits UDP traffic The default denies all UDP traffic.The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.

	Command	Purpose
Step 4	**permit all** **Example:** `switch(config-rbacl)# permit all`	Permits all traffic.
	permit icmp **Example:** `switch(config-rbacl)# permit icmp`	Permits ICMP traffic.
	permit igmp **Example:** `switch(config-rbacl)# permit igmp`	Permits IGMP traffic.
	permit ip **Example:** `switch(config-rbacl)# permit ip`	Permits IP traffic.
	permit tcp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1* *port-number2*}] **Example:** `switch(config-rbacl)# permit tcp`	Permits TCP traffic. The default permits all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535. The *port-number2* argument value must be greater than the *port-number1* argument value.
	permit udp [{**dest** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1* *port-number2*}] **Example:** `switch(config-rbacl)# permit udp dest ne 2000`	Permits UDP traffic. The default permits all UDP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535. The *port-number2* argument value must be greater than the *port-number1* argument value.

Step 5	**exit**
	
	Example:
	`switch(config-rbacl)# exit`
	`switch(config)#`
		Exits role-based access-list configuration mode.

Step 6	**cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name* **Example:** `switch(config)# cts role-based sgt 3 dgt 10 access-list MySGACL`	Maps the SGT values to the SGACL. The sgt-value and dgt-value arguments range from 0 to 65520. **Note** You must create the SGACL before you can map SGTs to it.
Step 7	**show cts role-based access-list** **Example:** `switch(config)# show cts role-based access-list`	(Optional) Displays the Cisco TrustSec SGACL configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Displaying the Downloaded SGACL Policies

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **show cts role-based access-list**

DETAILED STEPS

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

Refreshing the Downloaded SGACL Policies

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **cts refresh role-based-policy**
2. **show cts role-based policy**

DETAILED STEPS

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

Manually Configuring SXP

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

This section includes the following topics:

Cisco TrustSec SXP Configuration Process

Follow these steps to manually configure Cisco TrustSec SXP:

Step 1 Enable the Cisco TrustSec feature (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Step 2 Enable SGACL policy enforcement on the VRF (see the "Enabling SGACL Policy Enforcement on VRFs" section on page 9-31).

Step 3 Enable Cisco TrustSec SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

Step 4 Configure SXP peer connections (see the "Configuring Cisco TrustSec SXP Peer Connections" section on page 9-40).

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

Enabling Cisco TrustSec SXP

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

SUMMARY STEPS

1. **config t**
2. **cts sxp enable**
3. **exit**
4. **show cts sxp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

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

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

Configuring Cisco TrustSec SXP Peer Connections

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

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

Ensure that you enabled RBACL policy enforcement in the VRF (see the "Enabling SGACL Policy Enforcement on VRFs" section on page 9-31).

SUMMARY STEPS

1. **config t**
2. **cts sxp connection peer** *peer-ipv4-addr* [**source** *src-ipv4-addr*] **password** {**default** | **none** | **required** *password*} **mode** {**speaker** | **listener**} [**vrf** *vrf-name*]
3. **exit**
4. **show cts sxp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

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

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

Configuring the Default SXP Password

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

SUMMARY STEPS

1. **config t**
2. **cts sxp default password** *password*
3. **exit**
4. **show cts sxp**
5. **show running-config cts**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**cts sxp default password** *password* **Example:** switch(config)# cts sxp default password A2Q3d4F5	Configures the SXP default password.
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits configuration mode.
Step 4	**show cts sxp** **Example:** switch# show cts sxp	(Optional) Displays the SXP configuration.

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

Configuring the Default SXP Source IP Address

The NX-OS software uses default source IP address in all new TCP connections where a source IP address is not specified. There is no effect on existing TCP connections when you configure the default SXP source IP address.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

SUMMARY STEPS

1. **config t**
2. **cts sxp default source-ip** *src-ip-addr*
3. **exit**
4. **show cts sxp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

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

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

Changing the SXP Reconcile Period

After a peer terminates an SXP connection, an internal hold-down timer starts. If the peer reconnects before the internal hold-down timer expires, the SXP reconcile period timer starts. While the SXP reconcile period timer is active, the NX-OS software retains the SGT mapping entries learned from the previous connection and removes invalid entries. The default value is 120 seconds (2 minutes). Setting the SXP reconcile period to 0 seconds disables the timer and causes all entries from the previous connection to be removed.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

SUMMARY STEPS

1. **config t**
2. **cts sxp reconcile-period** *seconds*
3. **exit**
4. **show cts sxp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**cts sxp reconcile-period** *seconds* **Example:** `switch(config)# cts sxp reconcile-period 180`	Changes the SXP reconcile timer. The default value is 120 seconds (2 minutes). The range is from 0 to 64000.

	Command	Purpose

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

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

Changing the SXP Retry Period

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that you enabled Cisco TrustSec (see the "Enabling the Cisco TrustSec Feature" section on page 9-12).

Ensure that you enabled SXP (see the "Enabling Cisco TrustSec SXP" section on page 9-40).

SUMMARY STEPS

1. **config t**
2. **cts sxp retry-period** *seconds*
3. **exit**
4. **show cts sxp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

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

	Command	Purpose

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

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

Verifying Cisco TrustSec Configuration

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

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

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Cisco TrustSec Configurations

This sections includes the following topics:

- Enabling Cisco TrustSec, page 9-48
- Configuring AAA for Cisco TrustSec on a Seed NX-OS Device, page 9-48
- Enabling Cisco TrustSec Authentication on an Interface, page 9-48

Enabling Cisco TrustSec

The following example shows how to enable Cisco TrustSec:

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

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

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

```
radius-server host 10.10.1.1 key Cisco123 pac
aaa group server radius Rad1
  server 10.10.1.1
  use-vrf management
aaa authentication dot1x default group Rad1
aaa authorization cts default group Rad1
```

Enabling Cisco TrustSec Authentication on an Interface

The following example shows how to enable Cisco TrustSec authentication with a clear text password on an interface:

```
interface ethernet 2/1
  cts dot1x
shutdown
no shutdown
```

The following example shows how to enable Cisco TrustSec authentication with a clear text password on an interface:

```
interface ethernet 2/1
  cts dot1x
shutdown
no shutdown
```

Configuring Cisco TrustSec Authentication in Manual Mode

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

```
interface ethernet 2/1
  cts manual
    sap pmk abcdef modelist gmac
    policy static sgt 0x20
interface ethernet 2/2
  cts manual
    policy dynamic identity device2
```

Configuring Cisco TrustSec Role-Based Policy Enforcement for the default VRF

The following example shows how to enable Cisco TrustSec role-based policy enforcement for the default VRF:

```
cts role-based enforcement
```

Configuring Cisco TrustSec Role-Based Policy Enforcement for a Nondefault VRF

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

```
vrf context test
  cts role-based enforcement
```

Configuring Cisco TrustSec Role-Based Policy Enforcement for a VLAN

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

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

Configuring IPv4 Address to SGACL SGT Mapping for the Default VRF

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

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

Configuring IPv4 Address to SGACL SGT Mapping for a Nondefault VRF

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

```
vrf context test
```

```
  cts role-based sgt-map 30.1.1.1 30
```

Configuring IPv4 Address to SGACL SGT Mapping for a VLAN

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

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

Manually Configuring Cisco TrustSec SGACLs

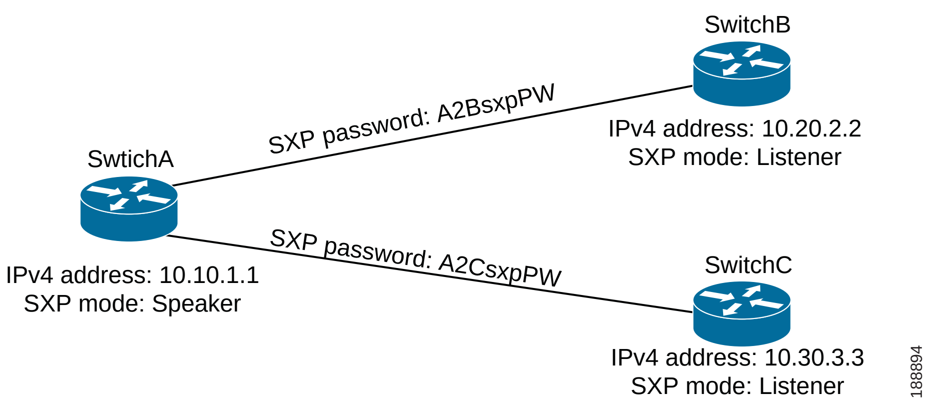
The following example shows how to manually configure Cisco TrustSec SGACLs:

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

Manually Configuring SXP Peer Connections

Figure 9-7 shows an example of SXP peer connections over the default VRF.

Figure 9-7 Example SXP Peer Connections



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

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

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

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

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

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

Default Settings

Table 9-1 lists the default settings for Cisco TrustSec parameters.

Table 9-1 Default Cisco TrustSec Parameters

Parameters	Default
Cisco TrustSec	Disabled.
SXP	Disabled.
SXP default password	None.
SXP reconcile period	120 seconds (2 minutes).
SXP retry period	60 seconds (1 minute).
Caching	Disabled.

Additional References

For additional information related to implementing Cisco TrustSec, see the following sections:

- Related Documents, page 9-51

Related Documents

Related Topic	Document Title
Cisco Secure ACS	Cisco Secure Access Control Server Engine Solution documentation
Command Reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
802.1X	Chapter 7, “Configuring 802.1X”

Feature History for Cisco TrustSec

Table 9-2 lists the release history for this feature.

Table 9-2 Feature History for IP ACLs

Feature Name	Releases	Feature Information
SGT propagation	4.0(3)	You can disable security group tag (SGT) propagation on Layer 2 Cisco TrustSec interfaces.
Cisco TrustSec manual configuration	4.0(3)	You can configure SAP for Cisco TrustSec manual mode to use 802.1X.
Cisco TrustSec	4.0(1)	This feature was introduced.

CHAPTER 10

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 ACLs.

This chapter includes the following sections:

Information 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 default rule. The device continues processing packets that are permitted and drops packets that are denied. For more information, see the "Implicit Rules" section on page 10-6.

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.

This section includes the following topics:

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.
- MAC ACLs—The device applies MAC ACLs only to non-IP traffic. For more information, see the "Information About MAC ACLs" section on page 11-1.
- Security-group ACLs (SGACLs)—The device applies SGACLs to traffic tagged by Cisco TrustSec. For more information, see Chapter 9, "Configuring Cisco TrustSec."

IP and MAC ACLs have the following three types of applications:

- Port ACL—Filters Layer 2 traffic
- Router ACL—Filters Layer 3 traffic
- VLAN ACL—Filters VLAN traffic

Table 10-1 summarizes the applications for security ACLs.

Table 10-1 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 • MAC ACLs
Router ACL	• VLAN interfaces (sometimes referred to as switched virtual interfaces or SVIs) **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.0*. • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	• IPv4 ACLs **Note** MAC ACLs are not supported on Layer 3 interfaces.
VLAN ACL	• VLANs For more information about VLAN ACLs, see Chapter 12, "Configuring VLAN ACLs."	• IPv4 ACLs • 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. SGACL
5. Egress router ACL
6. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs. Figure 10-1 shows the order in which the device applies ACLs.

Figure 10-1 Order of ACL Application

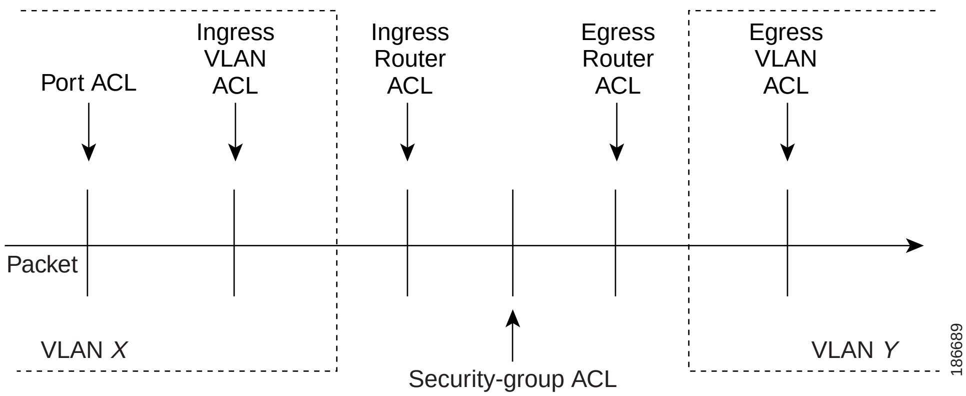
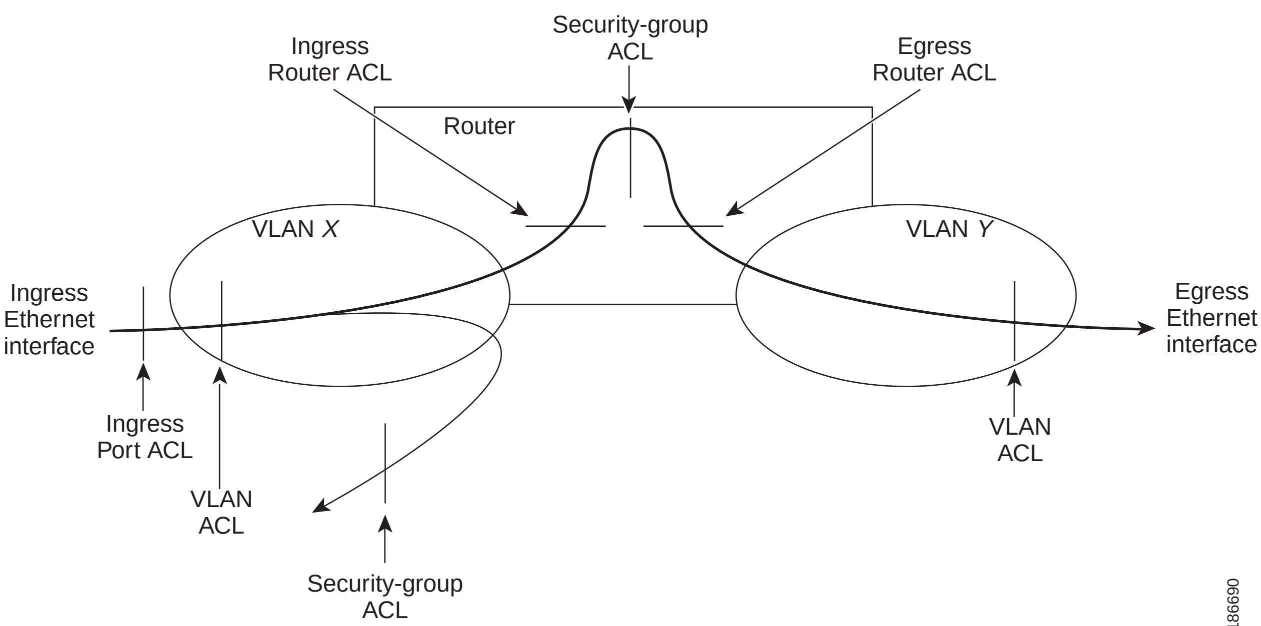


Figure 10-2 shows where the device applies ACLs, depending upon the type of ACL. The red path indicates a packet sent to a destination on a different interface than its source. The blue path indicates a packet that is bridged within its VLAN.

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.

For more information about SGACLs, see Chapter 9, "Configuring Cisco TrustSec."

Figure 10-2 ACLs and Packet Flow



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 use object groups when you configure rules. For more information, see the "Policy-Based ACLs" section on page 10-9.

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. For information about every option, see the applicable **permit** and **deny** commands in the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

This section includes the following topics:

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 or MAC ACLs. For information about specifying source and destination, see the applicable **permit** and **deny** commands in the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Protocols

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

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

For a list of the protocols that each type of ACL supports by name, see the applicable **permit** and **deny** commands in the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Implicit Rules

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 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
- MAC ACLs support the following additional filtering options:
  - Layer 3 protocol
  - VLAN ID
  - Class of Service (CoS)

For information about all filtering options available in rules, see the applicable **permit** and **deny** commands in the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

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). Cisco Nexus 7000-series devices support 104 LOUs.

The LOU usage for each type of operator is as follows:

- eq—Is never stored in an LOU
- gt—Uses 1/2 LOU
- lt—Uses 1/2 LOU
- neq—Uses 1/2 LOU
- range—Uses 1 LOU

The following guidelines determine when the devices store operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples “gt 10” and “gt 11” would be stored separately in half an LOU each. The couples “gt 10” and “lt 10” would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple “gt 10” to a source port and another rule applies a “gt 10” couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a “gt 10” couple would not result in further LOU usage.

Logging

You can enable the device to create an informational log message for packets that match a rule. The log message contains the following information about the packet:

- Protocol
- Status of whether the packet is a TCP, UDP, or ICMP packet, or if the packet is only a numbered packet.
- Source and destination address
- Source and destination port numbers, if applicable

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 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 ACL 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 a 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 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses or ports from the source or destination of rules. For example, if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the device expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the I/O module when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to port, router, and VLAN ACLs:

- IPv4 address object groups—Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.
- Protocol port object groups—Can be used with IPv4 TCP and UDP rules to specify source or destination ports. When you use the **permit** or **deny** command to configure a rule, the **portgroup** keyword allows you to specify an object group for the source or destination.

Statistics

The device can maintain global statistics for each rule that you configure in IPv4 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.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

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. For more information, see the "Implicit Rules" section on page 10-6.

For information about displaying IP ACL statistics, see the "Displaying and Clearing IP ACL Statistics" section on page 10-20. For information about displaying MAC ACL statistics, see the "Displaying and Clearing MAC ACL Statistics" section on page 11-8.

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. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Virtualization Support

The following information applies to IP and MAC ACLs used in Virtual Device Contexts (VDCs):

- ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The device does not limit ACLs or rules on a per-VDC basis.

Licensing Requirements for IP ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	IP ACLs require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

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

IP ACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the *CCisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.
- In most cases, ACL processing for IP packets are processed on the I/O modules. In some circumstances, processing occurs on the supervisor module, which is slower than the processing that occurs on I/O modules. Packets are processed on the supervisor module in the following circumstances:
  - Management interface traffic is always processed on the supervisor module.
  - IP packets exiting a Layer 3 interface that has an egress ACL with a large number of rules may be sent to the supervisor module.
- When you apply an ACL that uses time ranges, the device updates the ACL entries on the affected I/O modules whenever a time range referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally. For more information about VLAN interfaces, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.0*.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

Configuring IP ACLs

This section includes the following topics:

Creating an IP ACL

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *name*
3. [*sequence-number*] {**permit | deny**} *protocol source destination*
4. **statistics per-entry**
5. **show ip access-lists** *name*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	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.

	Command	Purpose
Step 3	`[`*`sequence-number`*`] {`**`permit`**` | `**`deny`**`}` *`protocol source destination`* **`Example:`** `switch(config-acl)# permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 4	**`statistics per-entry`** **`Example:`** `switch(config-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	**`show ip access-lists`** *`name`* **`Example:`** `switch(config-acl)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 6	**`copy running-config startup-config`** **`Example:`** `switch(config-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing an IP ACL

You can add and remove rules in an existing IPv4 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and 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. For more information, see the "Changing Sequence Numbers in an IP ACL" section on page 10-15.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

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

DETAILED STEPS

	Command	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)#`
		Enters IP ACL configuration mode for the ACL that you specify by name.

Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `switch(config-acl)# 100 permit ip 192.168.2.0/24 any`	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 4	**no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} **Example:** `switch(config-acl)# no 80`	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 5	[**no**] **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	**show ip access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	(Optional) 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 are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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 with the **summary** keyword to find the interfaces that an IP ACL is configured on.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

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

Step 2	**`no ip 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	**`show ip access-list`** *`name`* **`summary`** **Example:** `switch(config)# show ip access-lists acl-01 summary`	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**`copy running-config startup-config`** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

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

Step 2	**`resequence ip 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* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	**`show ip access-lists`** *`name`* **Example:** `switch(config)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 4	**`copy running-config startup-config`** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

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

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

BEFORE YOU BEGIN

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application. For more information, see the "Creating an IP ACL" section on page 10-12 or the "Changing an IP ACL" section on page 10-13.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*[**.***number*]

   interface port-channel *channel-number*[**.***number*]

   interface tunnel *tunnel-number*

   interface vlan *vlan-ID*

   interface mgmt *port*
3. **ip access-group** *access-list* {**in** | **out**}
4. **show running-config aclmgr**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

	Command	Purpose

Step 2	**interface ethernet** *slot*/*port*[.*number*]
	
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Enters interface configuration mode for a Layer 2 or Layer 3 physical interface. To enter configuration mode for a Layer 3 subinterface, specify the *number* argument.

	interface port-channel *channel-number*[.*number*]
	
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		Enters interface configuration mode for a port channel. To enter configuration mode for a Layer 3 port-channel interface, specify the *number* argument.

	interface tunnel *tunnel-number*
	
	Example:
	`switch(config)# interface tunnel 13`
	`switch(config-if)#`
		Enters interface configuration mode for a tunnel.

	interface vlan *vlan-ID*
	
	Example:
	`switch(config)# interface vlan 11`
	`switch(config-if)#`
		Enters interface configuration mode for a VLAN interface.

	interface mgmt *port*
	
	Example:
	`switch(config)# interface mgmt 0`
	`switch(config-if)#`
		Enters interface configuration mode for a management port.

Step 3	**ip access-group** *access-list* {**in** | **out**} **Example:** `switch(config-if)# ip access-group acl-120 out`	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	**show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a Port ACL

You can apply an IPv4 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.

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. For more information, see the "Creating an IP ACL" section on page 10-12 or the "Changing an IP ACL" section on page 10-13.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
   interface port-channel *channel-number*
3. **ip port access-group** *access-list* **in**
4. **show running-config aclmgr**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**interface ethernet** *slot*/*port* **Example:** switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode for a Layer 2 or Layer 3 physical interface.
	interface port-channel *channel-number* **Example:** switch(config)# interface port-channel 5 switch(config-if)#	Enters interface configuration mode for a port channel.
Step 3	**ip port access-group** *access-list* **in** **Example:** switch(config-if)# ip port access-group acl-l2-marketing-group in	Applies an IPv4 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	**show running-config aclmgr** **Example:** switch(config-if)# show running-config aclmgr	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config** **Example:** switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL. For information about how to create a VACL using an IPv4 ACL, see the "Creating or Changing a VACL" section on page 12-3.

Verifying IP ACL Configurations

To display IP ACL configuration information, use one of the following commands:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists	Displays the IPv4 ACL configuration.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying and Clearing IP ACL Statistics

To display or clear IP ACL statistics, use one of the following commands:

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

For detailed information about these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration 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
```

Configuring Object Groups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL rules.

This section includes the following topics:

Session Manager Support for Object Groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip address** *name*
3. [*sequence-number*] {**host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*}

   no {*sequence-number* | **host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*}
4. **show object-group** *name*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**object-group ip address** *name*
	
	Example:
	`switch(config)# object-group ip address ipv4-addr-group-13`
	`switch(config-ipaddr-ogroup)#`
		Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.

	Command	Purpose
Step 3	[*sequence-number*] {**host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*} **Example:** `switch(config-ipaddr-ogroup)# host 10.99.32.6`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command to specify a network of hosts.
	no [*sequence-number* | **host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*} **Example:** `switch(config-ipaddr-ogroup)# no host 10.99.32.6`	Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 4	**show object-group** *name* **Example:** `switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13`	(Optional) Displays the object group configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-ipaddr-ogroup)# copy running-config startup-config`	(Optional) 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.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip port** *name*
3. [*sequence-number*] *operator port-number* [*port-number*]

   no {*sequence-number* | *operator port-number* [*port-number*]}
4. **show object-group** *name*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	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 address 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 the port number that you specify only. • **gt**—Matches port numbers that are greater than (and not equal to) the port number that you specify. • **lt**—Matches port numbers that are less than (and not equal to) the port number that you specify. • **neq**—Matches all port numbers except for the port number that you specify. • **range**—Matches the range of port number between and including the two port numbers that you specify. **Note** The **range** command is the only operator command that requires two *port-number* arguments.
	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 4	**show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	(Optional) Displays the object group configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing an Object Group

You can remove an IPv4 address object group or a protocol port object group.

SUMMARY STEPS

1. **configure terminal**
2. **no object-group** {**ip address** | **ip port**} *name*
3. **show object-group**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no object-group** {**ip address** | **ip port**} *name* **Example:** `switch(config)# no object-group ip address ipv4-addr-group-A7`	Removes the object group that you specified.
Step 3	**show object-group** **Example:** `switch(config)# show object-group`	(Optional) Displays all object groups. The removed object group should not appear.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying Object-Group Configurations

To display object-group configuration information, use one of the following commands:

Command	Purpose
show object-group	Displays the object-group configuration
show running-config aclmgr	Displays ACL configuration, including object groups.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Configuring Time Ranges

This section includes the following topics:

Session Manager Support for Time Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Creating a Time Range

You can create a time range on the device and add rules to it.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*

   [*sequence-number*] **periodic** [*list-of-weekdays*] *time* **to** *time*

   [*sequence-number*] **absolute start** *time date* [**end** *time date*]

   [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*
4. **show time-range** *name*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

	Command	Purpose

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.

Step 3	[*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.
	[*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.
	[*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2008`	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.
	[*sequence-number*] **absolute** [start *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	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 4	**show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	(Optional) Displays the time-range configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	(Optional) 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. For more information, see the "Changing Sequence Numbers in a Time Range" section on page 10-29.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*

   [*sequence-number*] **periodic** [*list-of-weekdays*] *time* **to** *time*

   [*sequence-number*] **absolute start** *time date* [**end** *time date*]

   [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*

   no {*sequence-number* | **periodic** *arguments* . . . | **absolute** *arguments* . . .}
4. **show time-range** *name*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	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	[*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.
	[*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.
	[*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2008`	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.
	[*sequence-number*] **absolute** [**start** *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	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.
	no {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*} **Example:** `switch(config-time-range)# no 80`	Removes the specified rule from the time range.
Step 4	**show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	(Optional) Displays the time-range configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a Time Range

You can remove a time range from the device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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.

SUMMARY STEPS

1. **configure terminal**
2. **no time-range** *name*
3. **show time-range**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	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	**show time-range** **Example:** `switch(config-time-range)# show time-range`	(Optional) Displays configuration for all time ranges. The removed time range should not appear.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **resequence time-range** *name starting-sequence-number increment*
3. **show time-range** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	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	**`show time-range`** *`name`* **Example:** `switch(config)# show time-range daily-workhours`	(Optional) Displays the time-range configuration.
Step 4	**`copy running-config startup-config`** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying Time-Range Configurations

To display time-range configuration information, use one of the following commands:

Command	Purpose
show time-range	Displays the time-range configuration
show running-config aclmgr	Displays ACL configuration, including all time ranges.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Default Settings

Table 10-2 lists the default settings for IP ACL parameters.

Table 10-2 Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs (see the "Implicit Rules" section on page 10-6)
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default

Additional References

For additional information related to implementing IP ACLs, see the following sections:

- Related Documents, page 10-31
- Standards, page 10-31

Related Documents

Related Topic	Document Title
Concepts about VACLs	*Information About VLAN ACLs, page 12-1*
IP ACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
Object group commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
Time range commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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.	—

Feature History for IP ACLs

Table 10-3 lists the release history for this feature.

Table 10-3 Feature History for IP ACLs

Feature Name	Releases	Feature Information
Statistics	4.0(3)	The name of the **statistics** command was changed to **statistics per-entry**.
IP ACLs	4.0(1)	This feature was introduced.

CHAPTER 11

Configuring MAC ACLs

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

This chapter includes the following sections:

Information About MAC ACLs

MAC ACLs are ACLs that filter traffic using information in the Layer 2 header of each packet. MAC ACLs share many fundamental concepts with IP ACLs, including support for virtualization. For information about these shared concepts, see the "Information About ACLs" section on page 10-1.

Licensing Requirements for MAC ACLs

The following 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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for MAC ACLs

MAC ACLs have the following prerequisites:

- You must be familiar with MAC addressing and non-IP protocols to configure MAC ACLs.
- You must be familiar with the concepts in the "Information About ACLs" section on page 10-1.

Guidelines and Limitations

MAC ACLs have the following configuration guidelines and limitations:

- MAC ACLs apply to ingress traffic only.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

Configuring MAC ACLs

This section includes the following topics:

Creating a MAC ACL

You can create a MAC ACL and add rules to it.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **mac access-list** *name*
3. {**permit** | **deny**} *source destination protocol*
4. **statistics per-entry**
5. **show mac access-lists** *name*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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)# permit 00c0.4f00.0000 0000.00ff.ffff any`	Creates a rule in the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 4	**statistics per-entry** **Example:** `switch(config-mac-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	**show mac access-lists** *name* **Example:** `switch(config-mac-acl)# show mac access-lists acl-mac-01`	(Optional) Displays the MAC ACL configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-mac-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing a MAC ACL

In an existing MAC ACL, you can add and remove rules. 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. For more information, see the "Changing Sequence Numbers in a MAC ACL" section on page 11-6.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **mac access-list** *name*

3. [*sequence-number*] {**permit** | **deny**} *source destination protocol*
4. **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}
5. [**no**] **statistics per-entry**
6. **show mac access-lists** *name*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	[*sequence-number*] {**permit** | **deny**} *source destination protocol* **Example:** `switch(config-mac-acl)# 100 permit mac 00c0.4f00.00 0000.00ff.ffff any`	(Optional) Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 4	**no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*} **Example:** `switch(config-mac-acl)# no 80`	(Optional) Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 5	[**no**] **statistics per-entry** **Example:** `switch(config-mac-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	**show mac access-lists** *name* **Example:** `switch(config-mac-acl)# show mac access-lists acl-mac-01`	(Optional) Displays the MAC ACL configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-mac-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a MAC ACL

You can remove a MAC ACL from the device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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

SUMMARY STEPS

1. **config t**
2. **no mac access-list** *name*
3. **show mac access-lists** *name* **summary**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show mac access-lists** *name* **summary** **Example:** `switch(config)# show mac access-lists acl-mac-01 summary`	(Optional) Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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. For more information, see the "About Rules" section on page 10-5.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **resequence mac access-list** *name starting-sequence-number increment*
3. **show mac access-lists** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show mac access-lists** *name* **Example:** `switch(config)# show mac access-lists acl-mac-01`	(Optional) Displays the MAC ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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 interfaces
- Layer 3 interfaces

- 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. For more information about configuring MAC ACLs, see the "Configuring MAC ACLs" section on page 11-2.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
   interface port-channel *channel-number*
3. **mac port access-group** *access-list*
4. **show running-config aclmgr**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 for a Layer 2 or Layer 3 interface.

	interface port-channel *channel-number*
	
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		Enters interface configuration mode for a 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	**show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	(Optional) Displays ACL configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying a MAC ACL as a VACL

You can apply a MAC ACL as a VACL. For information about how to create a VACL using a MAC ACL, see the "Creating or Changing a VACL" section on page 12-3.

Verifying MAC ACL Configurations

To display MAC ACL configuration information, use one of the following commands:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration
show running-config aclmgr	Displays the ACL configuration, including MAC ACLs and the interfaces that ACLs are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying and Clearing MAC ACL Statistics

Use the **show mac access-lists** command to display statistics about a MAC ACL, including the number of packets that have matched each rule.

To display or clear MAC ACL statistics, use one of the following commands:

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

For detailed information about these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration 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
interface ethernet 2/1
  mac port access-group acl-mac-01
```

Default Settings

Table 11-1 lists the default settings for MAC ACL parameters.

Table 11-1 Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default
ACL rules	Implicit rules apply to all ACLs (see the "Implicit Rules" section on page 10-6)

Additional References

For additional information related to implementing MAC ACLs, see the following sections:

- Related Documents, page 11-9
- Standards, page 11-9

Related Documents

Related Topic	Document Title
Concepts about ACLs	*Information About ACLs, page 10-1*
MAC ACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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.	—

CHAPTER 12

Configuring VLAN ACLs

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

This chapter includes the following sections:

Information About VLAN ACLs

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

For more information about the types and applications of ACLs, see the "Information About ACLs" section on page 10-1.

This section includes the following topics:

VACLs and Access Maps

VACLs use access maps to link an IP ACL or a MAC ACL to an action. The device takes the configured action on packets permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.
- 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.

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.

For information about displaying VACL statistics, see the "Displaying and Clearing VACL Statistics" section on page 12-8.

Session Manager Support

Session Manager supports the configuration of VACLs. 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. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Virtualization Support

The following information applies to VACLs used in Virtual Device Contexts (VDCs):

- ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The device does not limit ACLs or rules on a per-VDC basis.

Licensing Requirements for VACLs

The following 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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for VACLs

VACLs have the following prerequisites:

- You must be familiar with VLANs to configure VACLs.
- You must be familiar with the concepts in the "Information About ACLs" section on page 10-1.

Guidelines and Limitations

VACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configurations using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.
- ACL statistics are not supported if the DHCP snooping feature is enabled.
- See the "Information About ACLs" section on page 10-1 section for more information about ACLs.

Configuring VACLs

This section includes the following topics:

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP or MAC ACL with an action to be applied to the matching traffic.

BEFORE YOU BEGIN

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. For more information about configuring IP ACLs, see the "Configuring IP ACLs" section on page 10-1. For more information about configuring MAC ACLs, see the "Configuring MAC ACLs" section on page 11-1.

SUMMARY STEPS

1. **config t**
2. **vlan access-map** *map-name*
3. **match ip address** *ip-access-list*
   match mac address *mac-access-list*
4. **action** {**drop** | **forward** | **redirect**}
5. [**no**] **statistics per-entry**
6. **show running-config aclmgr**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**`vlan access-map`** *`map-name`*
	
	Example:
	`switch(config)# vlan access-map acl-mac-map`
	`switch(config-access-map)#`
		Enters access map configuration mode for the access map specified.

Step 3	**`match ip address`** *`ip-access-list`* **Example:** `switch(config-access-map)# match mac address acl-ip-lab`	Specifies an IPv4 ACL for the map.
	`match mac address` *`mac-access-list`* **Example:** `switch(config-access-map)# match mac address acl-mac-01`	Specifies a MAC ACL for the map.

	Command	Purpose
Step 4	**action** {**drop** | **forward** | **redirect**} **Example:** `switch(config-access-map)# action forward`	Specifies the action that the device applies to traffic that matches the ACL. The **action** command supports many options. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.
Step 5	[**no**] **statistics per-entry** **Example:** `switch(config-access-map)# statistics per-entry`	(Optional) 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	**show running-config aclmgr** **Example:** `switch(config-access-map)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-access-map)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

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.

SUMMARY STEPS

1. **config t**
2. **no vlan access-map** *map-name*
3. **show running-config aclmgr**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no vlan access-map** *map-name* **Example:** `switch(config)# no vlan access-map acl-mac-map`	Removes the VLAN access map configuration for the specified access map.
Step 3	**show running-config aclmgr** **Example:** `switch(config)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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. For more information about creating VACLs, see the "Creating or Changing a VACL" section on page 12-3.

If you are unapplying a VACL, ensure that you are unapplying the correct VACL and that you understand how the VACL is currently applied. For more information about verifying the VACL configuration, see the "Verifying VACL Configuration" section on page 12-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **vlan filter** *map-name* **vlan-list** *list*
3. **show running-config aclmgr**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**show running-config aclmgr** **Example:** `switch(config)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying VACL Configuration

To display VACL configuration information, use one of the following commands:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including VACL-related configuration.
show vlan filter	Displays information about VACLs that are applied to a VLAN.
show vlan access-map	Displays information about VLAN access maps.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, use one of the following commands:

Command	Purpose
show vlan access-list	Displays the VACL configuration. If the VLAN access-map includes the **statistics per-entry** command, then 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 all VACLs or for a specific VACL.

For detailed information about these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration for VACL

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
```

Default Settings

Table 12-1 lists the default settings for VACL parameters.

Table 12-1 Default VACL Parameters

Parameters	Default
VACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs (see the "Implicit Rules" section on page 10-6)

Additional References

For additional information related to implementing IP ACLs, see the following sections:

- Related Documents, page 12-9
- Standards, page 12-9

Related Documents

Related Topic	Document Title
Concepts about ACLs	*Information About ACLs, page 10-1*
VACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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.	—

CHAPTER 13

Configuring Port Security

This chapter describes how to configure port security on NX-OS devices.

This chapter includes the following sections:

Information About Port Security

Port security allows you to configure Layer 2 interfaces that 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.

This section includes the following topics:

Secure MAC Address Learning

The process of securing a MAC address is called learning. The number of addresses that can be learned is restricted, as described in the "Secure MAC Address Maximums" section on page 13-3. For each interface that you enable port security on, the device can learn addresses by the static, dynamic, or sticky methods.

Static Method

The static learning method allows you to manually add or remove secure MAC addresses to the configuration of an interface.

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. For more information, see the "Removing a Static or a Sticky Secure MAC Address on an Interface" section on page 13-12.
- You configure the interface to act as a Layer 3 interface. For more information, see the "Port Type Changes" section on page 13-5.

Adding secure addresses by the static method is not affected by whether dynamic or sticky address learning is enabled.

Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device ages dynamic addresses and drops them once the age limit is reached, as described in the "Dynamic Address Aging" section on page 13-3.

Dynamic addresses do not persist through a device restart or through restarting the interface.

To remove a specific address learned by the dynamic method or to remove all addresses learned by the dynamic method on a specific interface, see the "Removing a Dynamic Secure MAC Address" section on page 13-13.

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 non-volatile 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.

The device does not age sticky secure MAC addresses.

To remove a specific address learned by the sticky method, see the "Removing a Static or a Sticky Secure MAC Address on an Interface" section on page 13-12.

Dynamic Address Aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 0 to 1440 minutes, where 0 disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The two methods of determining address age are as follows:

- Inactivity—The length of time after the device last received a packet from the address on the applicable interface.
- Absolute—The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Secure MAC Address Maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: dynamic, sticky, or static.

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 address 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 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 interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

  For an example of how VLAN and interface maximums interact, see the "Security Violations and Actions" section on page 13-4.

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. To remove dynamically learned addresses, see the "Removing a Dynamic Secure MAC Address" section on page 13-13. To remove addresses learned by the sticky or static methods, see the "Removing a Static or a Sticky Secure MAC Address on an Interface" section on page 13-12.

Security Violations and Actions

Port security triggers security violations when either of the two following events occur:

- Ingress traffic arrives at an interface from a nonsecure MAC address and learning the address would exceed the applicable maximum number of secure MAC addresses.

  When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

  - VLAN 1 has a maximum of 5 addresses
  - The interface has a maximum of 10 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 10 addresses on the interface and inbound traffic from an 11th address arrives at the interface.

- 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.

> **Note** After a secure MAC address is configured or learned on one secure port, the sequence of events that occurs when port security detects that secure MAC address on a different port in the same VLAN is known as a MAC move violation.

When a security violation occurs, the device takes the action specified by the port security configuration of the applicable 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 shut down** interface configuration commands.

- Restrict—Drops ingress traffic from any nonsecure MAC addresses. The device keeps a count of the number of dropped packets.
- Protect—Prevents violations from occurring. Address learning continues until the maximum number of MAC addresses on the interface is reached, after which the device disables learning on the interface and drops all ingress traffic from nonsecure MAC addresses.

If a violation occurs because ingress traffic from a secure MAC address arrives at a different interface than the interface on which the address is secure, the device applies the action on the interface that received the traffic.

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.
- Trunk ports—You can configure port security on interfaces that you have configured as Layer 2 trunk ports. VLAN maximums are not useful for access 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—Port security is not supported on Ethernet port channels.

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 or sticky method to the native trunk VLAN.
- Trunk port to access port—When you change a Layer 2 interface from a trunk port to an access port, the device drops all secure addresses learned by the dynamic method. It also moves all addresses learned by the sticky method on the native trunk VLAN to the access VLAN. The device drops secure addresses learned by the sticky method if they are not on the native trunk VLAN.
- Switched port to routed port—When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.
- Routed port to switched port—When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

802.1X and Port Security

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

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

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

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

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

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

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

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

Virtualization Support

Port security supports VDCs as follows:

- Port security is local to each VDC. You enable and configure port security on a per-VDC basis.
- Each VDC maintains secure MAC addresses separately.
- The device cannot issue a security violation when a secured MAC address in one VDC is seen on a protected interface in another VDC.

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 Cisco NX-OS device images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

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.

Guidelines and Limitations

When configuring port security, follow these guidelines:

- Port security does not support Ethernet port-channel interfaces or switched port analyzer (SPAN) destination ports.
- Port security does not depend upon other features.
- Port security can work with 802.1X, as described in the "802.1X and Port Security" section on page 13-5.

Configuring Port Security

This section includes the following topics:

Enabling or Disabling Port Security Globally

You can enable or disable port security globally on a device.

When you disable port security globally, all port security configuration is lost, including any statically configured secure MAC addresses and all dynamic or sticky secured MAC addresses.

BEFORE YOU BEGIN

By default, port security is disabled.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. [**no**] **feature port-security**
3. **show port-security**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **feature port-security** **Example:** `switch(config)# feature port-security`	Enables port security globally. The **no** option disables port security globally.
Step 3	**show port-security** **Example:** `switch(config)# show port-security`	Displays the status of port security.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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. For more information about dynamic learning of MAC addresses, see the "Secure MAC Address Learning" section on page 13-2.

Note You cannot enable port security on a routed interface.

BEFORE YOU BEGIN

By default, port security is disabled on all interfaces.

Enabling port security on an interface also enables dynamic MAC address learning. If you want to enable sticky MAC address learning, you must also complete the steps in the "Enabling or Disabling Sticky MAC Address Learning" section on page 13-9.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security, see the "Enabling or Disabling Port Security Globally" section on page 13-7.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **switchport**
4. [**no**] **switchport port-security**
5. **show running-config port-security**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the 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	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

By default, sticky MAC address learning is disabled.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled globally and on the interface that you are configuring. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security globally, see the "Enabling or Disabling Port Security Globally" section on page 13-7. To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 13-8.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **switchport**
4. [**no**] **switchport port-security mac-address sticky**
5. **show running-config port-security**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot/port*
	
	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	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

By default, no static secure MAC addresses are configured on an interface.

Determine if the interface maximum has been reached for secure MAC addresses (use the **show port-security** command). If needed, you can remove a secure MAC address (see the "Removing a Static or a Sticky Secure MAC Address on an Interface" section on page 13-12 or the "Removing a Dynamic Secure MAC Address" section on page 13-13) or you can change the maximum number of addresses on the interface (see the "Configuring a Maximum Number of MAC Addresses" section on page 13-13).

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled both globally and on the interface. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security globally, see the "Enabling or Disabling Port Security Globally" section on page 13-7. To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 13-8.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot/port*
	
	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	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a Static or a Sticky Secure MAC Address on an Interface

You can remove a static or a sticky secure MAC address on a Layer 2 interface.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security globally, see the "Enabling or Disabling Port Security Globally" section on page 13-7. To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 13-8.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **no switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	`interface` *`type slot/port`*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a secure static or sticky MAC address.

Step 3	`no switchport port-security mac-address` *`address`* **Example:** `switch(config-if)# no switchport port-security mac-address 0019.D2D0.00AE`	Removes the MAC address from port security on the current interface.

Step 4	`show running-config port-security`
	
	Example:
	`switch(config-if)# show running-config port-security`
		Displays the port security configuration.

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

Removing a Dynamic Secure MAC Address

You can remove dynamically learned, secure MAC addresses.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **clear port-security dynamic** {**interface ethernet** *slot/port* | **address** *address*} [**vlan** *vlan-ID*]
3. **show port-security address**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**clear port-security dynamic** {**interface ethernet** *slot/port* | **address** *address*} [**vlan** *vlan-ID*] **Example:** `switch(config)# clear port-security dynamic interface ethernet 2/1`	Removes dynamically learned, secure MAC addresses, as specified. If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify. If you use the **address** keyword, you remove the single, dynamically learned address that you specify. Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.

Step 3	`show port-security address`
	
	Example:
	`switch(config)# show port-security address`
		Displays secure MAC addresses.

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 is 4096 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 reduce the number of addresses learned by the sticky or static methods, see the "Removing a Static or a Sticky Secure MAC Address on an Interface" section on page 13-12. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

BEFORE YOU BEGIN

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security, see the "Enabling or Disabling Port Security Globally" section on page 13-7.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot*
3. [**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot*
	
	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 4096. 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.

	Command	Purpose
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

By default, the aging time is 0 minutes, which disables aging.

Absolute aging is the default aging type.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security, see the "Enabling or Disabling Port Security Globally" section on page 13-7.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot*
3. [**no**] **switchport port-security aging type** {**absolute** | **inactivity**}
4. [**no**] **switchport port-security aging time** *minutes*
5. **show running-config port-security**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot*
	
	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 MAC aging type and time.

	Command	Purpose
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	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

The default security action is to shut down the port on which the security violation occurs.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that port security is enabled. To verify the configuration, see the "Verifying the Port Security Configuration" section on page 13-17. To enable port security, see the "Enabling or Disabling Port Security Globally" section on page 13-7.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. [**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**}
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface** *type slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot* is the interface for which you want to configure the 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	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying the Port Security Configuration

To display the port security configuration information, use the following commands:

Command	Purpose
show running-config port-security	Displays the port security configuration
show port-security	Displays the port security status.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses. For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration 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
```

Default Settings

Table 13-1 lists the default settings for port security parameters.

Table 13-1 Default Port Security Parameters

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown

Additional References

For additional information related to implementing port security, see the following sections:

- Related Documents, page 13-18
- Standards, page 13-19
- MIBs, page 13-19

Related Documents

Related Topic	Document Title
Layer 2 switching	*Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.0*
Port security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

Standards

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

MIBs

NX-OS provides read-only SNMP support for port security.

MIBs	MIBs Link
• CISCO-PORT-SECURITY-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/nx-os/mibs

CHAPTER 14

Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping on an NX-OS device.

This chapter includes the following sections:

Information About DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

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

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

This section includes the following topics:

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.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command. For more information, see the "Clearing the DHCP Snooping Binding Database" section on page 14-17.

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 agent forwarding is distinct from the normal forwarding of an IP router, where IP datagrams are switched between networks somewhat transparently. By contrast, 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 (option82) 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.

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):

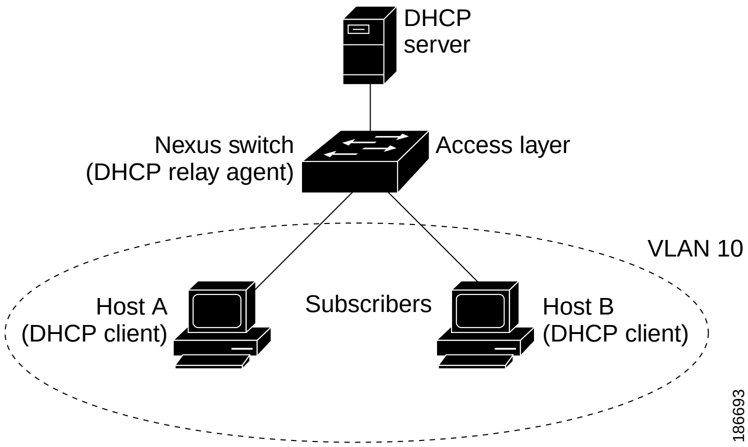
- The device receives a DHCP response packet (such as DHCPACK, DHCPNAK, or DHCPOFFER packet) on an untrusted interface.
- The device receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping 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.
- The device receives a DHCP packet that includes a relay agent IP address that is not 0.0.0.0.

DHCP Snooping Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, 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.

Figure 14-1 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.

Figure 14-1 DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable option 82 on the NX-OS device, the following sequence of events occurs:

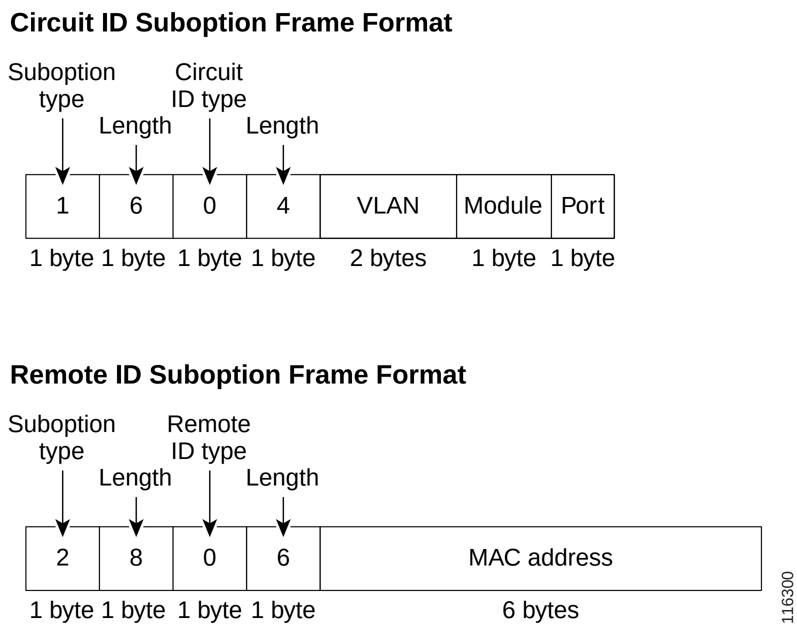
1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the NX-OS device receives the DHCP request, it adds the option-82 information in the packet. The option-82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
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.

If the previously described sequence of events occurs, the following values (see Figure 14-2) 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

Figure 14-2 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 14-2 Suboption Packet Formats



Virtualization Support for DHCP Snooping

The following information applies to DHCP snooping used in Virtual Device Contexts (VDCs):

- DHCP snooping binding databases are unique per VDC. Bindings in one VDC do not affect DHCP snooping in other VDCs.
- The system does not limit binding database size on a per-VDC basis.

Licensing Requirements for DHCP Snooping

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	DHCP snooping requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for DHCP Snooping

DHCP snooping has the following prerequisites:

- You must be familiar with DHCP to configure DHCP snooping.

Guidelines and Limitations

DHCP snooping has the following configuration guidelines and limitations:

- When you use the **feature dhcp** command to enable the DHCP snooping feature, there is a delay of approximately 30 seconds before the I/O modules receive DHCP snooping or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with DHCP snooping disabled to a configuration with DHCP snooping enabled. For example, if you use the Rollback feature to revert to a configuration that enables DHCP snooping, the I/O modules receive DHCP snooping and DAI configuration approximately 30 seconds after you complete the rollback.
- The DHCP snooping database can store 2000 bindings.
- DHCP snooping is not active until you enable the feature, enable DHCP snooping globally, and enable DHCP snooping on at least one VLAN.
- Before globally enabling DHCP snooping on the device, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.
- Access-control list (ACL) statistics are not supported if the DHCP snooping feature is enabled.

Configuring DHCP Snooping

This section includes the following topics:

Minimum DHCP Snooping Configuration

The minimum configuration for DHCP snooping is as follows:

Step 1 Enable the DHCP snooping feature. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

When the DHCP snooping feature is disabled, you cannot configure DHCP snooping.

Step 2 Enable DHCP snooping globally. For more information, see the "Enabling or Disabling DHCP Snooping Globally" section on page 14-8.

Step 3 Enable DHCP snooping on at least one VLAN. For more information, see the "Enabling or Disabling DHCP Snooping on a VLAN" section on page 14-9.

By default, DHCP snooping is disabled on all VLANs.

Step 4 Ensure that the DHCP server is connected to the device using a trusted interface. For more information, see the "Configuring an Interface as Trusted or Untrusted" section on page 14-12.

Step 5 (Optional) Enable the DHCP relay agent. For more information, see the "Enabling or Disabling the DHCP Relay Agent" section on page 14-13.

Step 6 (Optional) Configure an interface with the IP address of the DHCP server. For more information, see the "Configuring DHCP Server Addresses on an Interface" section on page 14-15. one of the following topics:

Enabling or Disabling the DHCP Snooping Feature

You can enable or disable the DHCP snooping feature on the device. By default, DHCP snooping is disabled.

BEFORE YOU BEGIN

If you disable the DHCP snooping feature, all DHCP snooping configuration is lost. If you want to turn off DHCP snooping and preserve the DHCP snooping configuration, disable DHCP globally. For more information, see the "Enabling or Disabling DHCP Snooping Globally" section on page 14-8.

SUMMARY STEPS

1. **config t**
2. [**no**] **feature dhcp**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **feature dhcp** **Example:** `switch(config)# feature dhcp`	Enables the DHCP snooping feature. The **no** option disables the DHCP snooping feature and erases all DHCP snooping configuration.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping Globally

You can enable or disable the DHCP snooping globally on the device.

BEFORE YOU BEGIN

By default, DHCP snooping is globally disabled.

Ensure that you have enabled the DHCP snooping feature. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

Globally disabling DHCP snooping stops the device from performing any DHCP snooping or relaying DHCP messages. It preserves DCHP snooping configuration.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** option disables DHCP snooping.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs.

BEFORE YOU BEGIN

By default, DHCP snooping is disabled on all VLANs.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping vlan** *vlan-list*
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:** `switch(config)# ip dhcp snooping vlan 100,200,250-252`	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP snooping on the VLANs specified.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

MAC address verification is enabled by default.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping verify mac-address**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	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** option disables MAC address verification.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

Note You must separately configure the DHCP relay agent to support option 82. For more information, see the "Enabling or Disabling Option 82 for the DHCP Relay Agent" section on page 14-14.

BEFORE YOU BEGIN

By default, the device does not include option-82 information in DHCP packets.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping information option**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 from DHCP packets. The **no** option disables the insertion and removal of option-82 information.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring an Interface as Trusted or Untrusted

You can configure whether an interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

BEFORE YOU BEGIN

By default, all interfaces are untrusted.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

Ensure that the interface is configured as a Layer 2 interface.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*

   interface port-channel *channel-number*
3. [**no**] **ip dhcp snooping trust**
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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, where *slot/port* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.

	interface port-channel *channel-number*
	
	Example:
	`switch(config)# interface port-channel 5`
	`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** option configures the port as an untrusted interface.
Step 4	**show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling the DHCP Relay Agent

You can enable or disable the DHCP relay agent.

BEFORE YOU BEGIN

By default, the DHCP relay agent is disabled.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **service dhcp**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **service dhcp** **Example:** `switch(config)# service dhcp`	Enables the DHCP relay agent. The **no** option disables the DHCP relay agent.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

By default, the DHCP relay agent does not include option-82 information in DHCP packets.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay information option**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **ip dhcp relay information option** **Example:** `switch(config)# ip dhcp relay information option`	Enables the DHCP relay agent to insert and remove option 82 information from the packets that it forwards. The **no** option disables this behavior.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring DHCP Server Addresses on an Interface

You can configure up to four 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 address specified. The relay agent forwards replies from all DHCP servers to the host that sent the request. In Cisco NX-OS Release 4.0.2 and earlier, you can configure only one DHCP server IP address on an interface.

BEFORE YOU BEGIN

By default, there is no DHCP server IP address configured on an interface.

Ensure that the DHCP server is correctly configured.

Determine the IP address for each DHCP server that you want to .

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*[.*number*]
   interface vlan *vlan-id*
   interface port-channel *channel-id*
3. **ip dhcp relay address** *IP-address*
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**interface ethernet** *slot*/*port*[.*number*]
	
	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.

	interface vlan *vlan-id*
	
	Example:
	`switch(config)# interface vlan 13`
	`switch(config-if)#`
		Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.

	interface port-channel *channel-id*
	
	Example:
	`switch(config)# interface port-channel 7`
	`switch(config-if)#`
		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.

Step 3	**ip dhcp relay address** *IP-address* **Example:** `switch(config-if)# ip dhcp relay address 10.132.7.120`	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. You can configure up to four addresses.
Step 4	**show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying DHCP Snooping Configuration

To display DHCP snooping configuration information, use the following commands:

Command	Purpose
show running-config dhcp	Displays the DHCP snooping configuration
show ip dhcp snooping	Displays general information about DHCP snooping.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying DHCP Bindings

Use the **show ip dhcp snooping binding** command to display the DHCP binding table. For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Clearing the DHCP Snooping Binding Database

You can remove all entries from the DHCP snooping binding database.

BEFORE YOU BEGIN

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **clear ip dhcp snooping binding**
2. **show ip dhcp snooping binding**

DETAILED STEPS

	Command	Purpose
Step 1	**`clear ip dhcp snooping binding`** **Example:** `switch# clear ip dhcp snooping binding`	Clears the DHCP snooping binding database.
Step 2	**`show ip dhcp snooping binding`** **Example:** `switch# ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Displaying DHCP Snooping Statistics

Use the **show ip dhcp snooping statistics** command to display DHCP snooping statistics. For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration for DHCP Snooping

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 info option

interface Ethernet2/5
  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 server IP address is 10.132.7.120:

```
feature dhcp
ip dhcp snoop
service dhcp
ip dhcp relay information option

interface Ethernet2/3
  ip dhcp relay address 10.132.7.120
```

Default Settings

Table 14-1 lists the default settings for DHCP snooping parameters.

Table 14-1 Default DHCP Snooping Parameters

Parameters	Default
DHCP snooping feature	Disabled
DHCP snooping globally enabled	No
DHCP snooping VLAN	None
DHCP snooping MAC address verification	Enabled
DHCP snooping option-82 support	Disabled
DHCP snooping trust	Untrusted
DHCP snooping relay agent	Disabled
DHCP snooping option-82 for relay agent	Disabled
DHCP server IP address	None

Additional References

For additional information related to implementing DHCP snooping, see the following sections:

- Related Documents, page 14-19
- Standards, page 14-19

Related Documents

Related Topic	Document Title
IP Source Guard	Information About IP Source Guard, page 16-1
Dynamic ARP Inspection	Information About DAI, page 15-1
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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)

Feature History for DHCP Snooping

Table 14-2 lists the release history for this feature.

Table 14-2 Feature History for DHCP Snooping

Feature Name	Releases	Feature Information
Multiple DHCP server support	4.0(3)	The number of DHCP server addresses that you can configure for each Layer 3 Ethernet interface increased from one to four.
DHCP snooping	4.0(1)	This feature was introduced.

CHAPTER 15

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on an NX-OS device.

This chapter includes the following sections:

Information About DAI

This section includes the following topics:

Understanding 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.

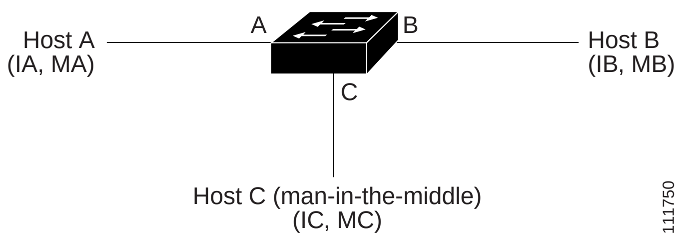
To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

Understanding 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 15-1 shows an example of ARP cache poisoning.

Figure 15-1 ARP Cache Poisoning



Hosts A, B, and C are connected to the device on interfaces A, B, and C, all of 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 the 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.

Understanding DAI and ARP Spoofing Attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, an 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 Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the device. It 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.

DAI can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses (see the "Applying ARP ACLs to VLANs for DAI Filtering" section on page 15-9). The device logs dropped packets (see the "Logging DAI Packets" section on page 15-5).

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 (see the "Enabling or Disabling Additional Validation" section on page 15-11).

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 as follows:

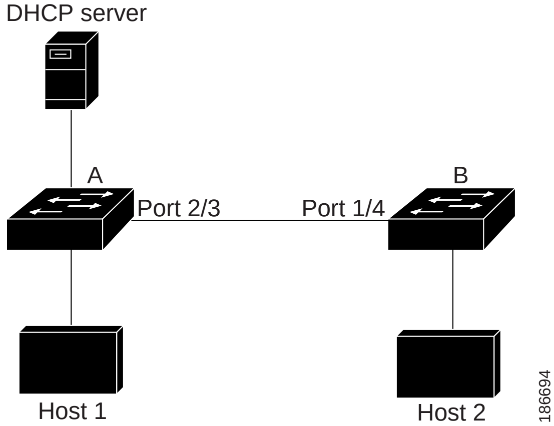
- Untrusted—Interfaces that are connected to hosts
- Trusted—Interfaces that are connected to devices

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. For information about configuring the trust state of an interface, see the "Configuring the DAI Trust State of a Layer 2 Interface" section on page 15-8.

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In Figure 15-2, assume 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.

Figure 15-2 ARP Packet Validation on a VLAN Enabled for DAI



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, then the guidelines for configuring the trust state of interfaces on a device running DAI becomes 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

To validate the bindings of packets from devices that are not running DAI, configure ARP ACLs on the device running DAI. When you cannot determine the bindings, isolate at Layer 3 the devices that run DAI from devices that do not run DAI. For configuration information, see the "Example 2: One Device Supports DAI" section on page 15-19.

Note Depending on your network setup, you may not be able to validate a given ARP packet on all devices in the VLAN.

Prioritizing ARP ACLs and DHCP Snooping Entries

By default, DAI filters DAI traffic by comparing DAI packets to IP-MAC address bindings in the DHCP snooping database.

When you apply an ARP ACL to traffic, the ARP ACLs take precedence over the default filtering behavior. The device first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the device denies the packet regardless of whether a valid IP-MAC binding exists in the DHCP snooping database.

Note VLAN ACLs (VACLs) take precedence over both ARP ACLs and DHCP snooping entries. For example, if you apply a VACL and an ARP ACL to a VLAN and you configured the VACL to act on ARP traffic, the device permits or denies ARP traffic as determined by the VACL, not the ARP ACL or DHCP snooping entries.

For information about configuring ARP ACLs, see the "Configuring ARP ACLs" section on page 15-21. For information about applying an ARP ACL, see the "Applying ARP ACLs to VLANs for DAI Filtering" section on page 15-9.

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, an NX-OS device logs only packets that DAI drops. For configuration information, see the "Configuring DAI Log Filtering" section on page 15-13.

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. For more information, see the "Configuring the DAI Logging Buffer Size" section on page 15-12.

Note NX-OS does not generate system messages about DAI packets that are logged.

Virtualization Support

The following information applies to DAI used in Virtual Device Contexts (VDCs):

- IP-MAC address bindings are unique per VDC.
- ARP ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The system does not limit ARP ACLs or rules on a per-VDC basis.

Licensing Requirements for DAI

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	DAI requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for DAI

You should be familiar with the following before you configure DAI:

- ARP
- DHCP snooping

Guidelines and Limitations

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.
- 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. If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, DHCP snooping needs only to be enabled. If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, DHCP snooping must configured on the same VLANs on which you configure DAI. For configuration information, see the "Configuring DHCP Snooping" section on page 14-6.
- When you use the **feature dhcp** command to enable the DHCP snooping feature, there is a delay of approximately 30 seconds before the I/O modules receive DHCP snooping or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with DHCP snooping disabled to a configuration with DHCP snooping enabled. For example, if you use the Rollback feature to revert to a configuration that enables DHCP snooping, the I/O modules receive DHCP snooping and DAI configuration approximately 30 seconds after you complete the rollback.
- When DHCP snooping is disabled or used in a non-DHCP environment, you should use ARP ACLs to permit or to deny packets.
- DAI is supported on access ports, trunk ports, port-channel ports, and private VLAN 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 retains the DAI trust state configuration of the port channel.
- When you change the trust state on the port channel, the device configures a new trust state on all the physical ports that comprise the channel.
- If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, ensure that the DHCP snooping feature is enabled and that you have configured the static IP-MAC address bindings. For configuration information, see the "Configuring DHCP Snooping" section on page 14-6.
- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is configured (see the "Configuring DHCP Snooping" section on page 14-6).

Configuring DAI

This section includes the following topics:

Enabling or Disabling DAI on VLANs

You can enable or disable DAI on VLANs.

BEFORE YOU BEGIN

By default, DAI is disabled on all VLANs.

If you are enabling DAI, ensure the following:

- DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.
- The VLANs on which you want to enable DAI are configured.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection vlan** *list*
3. **show ip arp inspection vlan** *list*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **ip arp inspection 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	**show ip arp inspection vlan** *list* **Example:** `switch(config)# show ip arp inspection vlan 13`	(Optional) Shows the DAI status for the specified list of VLANs.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

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, 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. For more information, see the "Configuring DAI Log Filtering" section on page 15-13.

BEFORE YOU BEGIN

By default, all interfaces are untrusted.

If you are enabling DAI, ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/number*
3. [**no**] **ip arp inspection trust**
4. **show ip arp inspection interface** *type slot/number*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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	**show ip arp inspection interface** *type slot***/***number* **Example:** `switch(config-if)# show ip arp inspection interface ethernet 2/1`	(Optional) Displays the trust state and the ARP packet rate for the specified interface.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying ARP ACLs to VLANs for DAI Filtering

You can apply an ARP ACL to one or more VLANs. The device permits packets only if the ACL permits them.

BEFORE YOU BEGIN

By default, no VLANs have an ARP ACL applied.

Ensure that the ARP ACL that you want to apply is correctly configured. For information about configuring an ARP ACL, see the "Configuring ARP ACLs" section on page 15-21.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection filter** *acl-name* **vlan** *list*
3. **show ip arp inspection vlan** *list*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ip arp inspection filter** *acl-name* **vlan** *list* **Example:** switch(config)# ip arp inspection filter arp-acl-01 vlan 100	Applies the ARP ACL to the list of VLANs, or if you use the **no** option, removes the ARP ACL from the list of VLANs.
Step 3	**show ip arp inspection vlan** *list* **Example:** switch(config)# show ip arp inspection vlan 100	(Optional) Shows the DAI status for the specified list of VLANs, including whether an ARP ACL is applied.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling DAI Error-Disabled Recovery

You can enable or disable DAI error-disabled recovery on a device.

BEFORE YOU BEGIN

By default, DAI error-disabled recovery is disabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **errdisable recovery cause arp-inspection**
3. **show running-config | include errdisable**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **errdisable recovery cause arp-inspection** **Example:** `switch(config)# errdisable recovery cause arp-inspection`	Enables DAI error-disabled recovery. The **no** option disables DAI error-disabled recovery.
Step 3	**show running-config | include errdisable** **Example:** `switch(config)# show running-config | include errdisable`	(Optional) Displays the errdisable configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling Additional Validation

You can enable or disable additional validation of ARP packets.

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.

BEFORE YOU BEGIN

By default, no additional validation of ARP packets is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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, or if you use the **no** option, disables additional DAI validation.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The additional validations do the following:

- **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 overrides 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 only, the **src-mac** and **dst-mac** validations are disabled when you enter the second command.

Configuring the DAI Logging Buffer Size

You can configure the DAI logging buffer size.

BEFORE YOU BEGIN

The default buffer size is 32 messages.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection log-buffer entries** *number*
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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 0 and 2048 messages.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

By default, the device logs DAI packets that are dropped.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**}
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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. The **no** option removes DAI log filtering.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

When configuring the DAI log filtering, follow these guidelines:

- By default, all denied packets are logged.
- **dhcp-bindings all**—Logs all packets that match DHCP bindings.
- **dhcp-bindings none**—Does not log packets that match DHCP bindings.
- **dhcp-bindings permit**—Logs DHCP-binding permitted packets.

Verifying the DAI Configuration

To display the DAI configuration information, use the following commands:

Command	Purpose
show running-config arp	Displays DAI configuration.
show ip arp inspection	Displays the status of DAI.
show ip arp inspection interface ethernet *slot/port*	Displays the trust state and ARP packet rate for a specific interface.
show ip arp inspection vlan *vlan-ID*	Displays the DAI configuration for a specific VLAN.
show arp access-lists	Displays ARP ACLs.
show ip arp inspection log	Displays the DAI log configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying and Clearing DAI Statistics

To display and clear DAI statistics, use the following commands:

Command	Purpose
show ip arp inspection statistics	Displays DAI statistics.
show arp ethernet *slot/port* **statistics**	Displays interface-specific DAI statistics.
clear ip arp inspection statistics	Clears DAI statistics.

For more information about these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configurations for DAI

This section includes these examples:

- Example 1: Two Devices Support DAI, page 15-15
- Example 2: One Device Supports DAI, page 15-19

Example 1: Two Devices Support DAI

This procedure shows how to configure DAI when two devices support this feature. Host 1 is connected to device A, and Host 2 is connected to device B as shown in Figure 15-2 on page 15-4. 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 the device B Ethernet interface 1/4.

Note

- 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. For configuration information, see Chapter 14, "Configuring DHCP Snooping."
- 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:

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# config t
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, 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 Jul 13 2008])
```

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:

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# config t
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.

```
witchB# 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 Jun 13 2008])
switchB#
```

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#
```

Example 2: One Device Supports DAI

This procedure shows how to configure DAI when device B shown in Figure 15-2 on page 15-4 does not support DAI or DHCP snooping.

If device B does not support DAI or DHCP snooping, configuring Ethernet interface 2/3 on device A as trusted creates a security hole because both device A and Host 1 could be attacked by either device B or Host 2.

To prevent this possibility, you must configure Ethernet interface 2/3 on device A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 1. If the IP address of Host 2 is not static, which would make it impossible to accurately configure the ARP ACL on device A, you must separate device A from device B at Layer 3 and use a router to route packets between them.

To set up an ARP ACL on device A, follow these steps:

Step 1 Configure the access list to permit the IP address 10.0.0.1 and the MAC address 0001.0001.0001, and verify the configuration.

```
switchA# config t
switchA(config)# arp access-list H2
switchA(config-arp-acl)# permit ip host 10.0.0.1 mac host 0001.0001.0001
switchA(config-arp-acl)# exit
switchA(config)# show arp access-lists H2

ARP access list H2
10 permit ip host 1.1.1.1 mac host 0001.0001.0001
switchA(config)#
```

Step 2 Apply the ACL to VLAN 1, and verify the configuration.

```
switchA(config)# ip arp inspection filter H2 vlan 1
switchA(config)# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

Vlan : 200
-----------
Configuration    : Enabled
Operation State  : Active
ACL Match/Static : H2 / No
```

Step 3 Configure Ethernet interface 2/3 as untrusted, and verify the configuration.

Note By default, the interface is untrusted.

```
switchA(config)# interface ethernet 2/3
switchA(config-if)# no ip arp inspection trust
switchA(config-if)# exit
switchA# show ip arp inspection interface ethernet 2/3
switchA#
```

The **show ip arp inspection interface** command has no output because the interface has the default configuration, which includes an untrusted state.

When Host 2 sends 5 ARP requests through Ethernet interface 2/3 on device A and a “get” is permitted by device A, the statistics are updated.

```
switchA# show ip arp inspection statistics vlan 1

Vlan : 1
-----------
ARP Req Forwarded  = 5
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#
```

Configuring ARP ACLs

This section includes the following topics:

Session Manager Support

Session Manager supports the configuration of ARP ACLs. This feature allows you to create a configuration session and verify your ARP ACL configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Creating an ARP ACL

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). ACL names can be repeated in different VDCs, so we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **arp access-list** *name*
3. [*sequence-number*] {**permit** | **deny**} **ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**]

   [*sequence-number*] {**permit** | **deny**} **request ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**]

   [*sequence-number*] {**permit** | **deny**} **response ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} {**any** | **host** *target-IP* | *target-IP target-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**any** | **host** *target-MAC* | *target-MAC target-MAC-mask*] [**log**]
4. **show arp access-lists** *acl-name*
5. **copy running-config startup-config**

Send document comments to nexus7k-docfeedback@cisco.com

DETAILED STEPS

	Command	Purpose

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

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

Step 3	[*sequence-number*] {**permit** | **deny**} **ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**] **Example:** `switch(config-arp-acl)# permit ip 192.168.2.0 0.0.0.255 mac 00C0.4F00.0000 ffff.ff00.0000`	Creates a rule that permits or denies any ARP message based upon the IP address and MAC address of the sender of the message. 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.
	[*sequence-number*] {**permit** | **deny**} **request ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**] **Example:** `switch(config-arp-acl)# permit request ip 192.168.102.0 0.0.0.255 mac any`	Creates a rule that permits or denies ARP request messages based upon the IP address and MAC address of the sender of the message. 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.
	[*sequence-number*] {**permit** | **deny**} **response ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} [**any** | **host** *target-IP* | *target-IP target-IP-mask*]] **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**any** | **host** *target-MAC* | *target-MAC target-MAC-mask*] [**log**] **Example:** `switch(config-arp-acl)# permit response ip host 192.168.202.32 any mac host 00C0.4FA9.BCF3 any`	Creates a rule that permits or denies ARP response messages based upon the IPv4 address and MAC address of the sender and the target of the message. 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.
Step 4	**show arp access-lists** *acl-name* **Example:** `switch(config-arp-acl)# show arp access-lists arp-acl-01`	(Optional) Shows the ARP ACL configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-arp-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing an ARP ACL

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

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see the "Changing Sequence Numbers in an ARP ACL" section on page 15-25.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). ACL names can be repeated in different VDCs, so we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **arp access-list** *name*
3. [*sequence-number*] {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data*
4. **no** {*sequence-number* | {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data*}
5. **show arp access-lists**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`arp access-list`** *`name`*
	
	`Example:`
	`switch(config)# arp access-list arp-acl-01`
	`switch(config-acl)#`
		Enters ARP ACL configuration mode for the ACL that you specify by name.

	Command	Purpose
Step 3	[*sequence-number*] {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data* **Example:** `switch(config-arp-acl)# 100 permit request ip 192.168.132.0 0.0.0.255 mac any`	(Optional) Creates a rule. For more information about the **permit** and **deny** commands, see the "Creating an ARP ACL" section on page 15-21. 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.
Step 4	**no** {*sequence-number* | {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data* **Example:** `switch(config-arp-acl)# no 80`	(Optional) Removes the rule that you specified from the ARP ACL. For more information about the **permit** and **deny** commands, see the "Creating an ARP ACL" section on page 15-21.
Step 5	**show arp access-lists** **Example:** `switch(config-arp-acl)# show arp access-lists`	Displays the ARP ACL configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-arp-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing an ARP ACL

You can remove an ARP ACL from the device.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). ACL names can be repeated in different VDCs, so we recommend that you confirm which VDC you are working in.

Ensure that you know whether the ACL is applied to a VLAN. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of VLANs where you have applied the ACL. Instead, the device considers the removed ACL to be empty.

SUMMARY STEPS

1. **config t**
2. **no arp access-list** *name*
3. **show arp access-lists**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no arp access-list** *name* **Example:** `switch(config)# no arp access-list arp-acl-01`	Removes the ARP ACL you specified by name from running configuration.
Step 3	**show arp access-lists** **Example:** `switch(config)# show arp access-lists`	Displays the ARP ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an ARP ACL

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command). ACL names can be repeated in different VDCs, so we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **config t**
2. **resequence arp access-list** *name starting-sequence-number increment*
3. **show arp access-lists** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**resequence arp access-list** *name starting-sequence-number increment*
	
	Example:
	`switch(config)# resequence arp access-list arp-acl-01 100 10`
	`switch(config)#`
		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.

Step 3	**show arp access-lists** *name* **Example:** `switch(config)# show arp access-lists arp-acl-01`	Displays the ARP ACL configuration for the ACL specified by the *name* argument.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying ARP ACL Configuration

To display ARP ACL configuration information, use one of the following commands:

Command	Purpose
show arp access-lists	Displays the ARP ACL configuration.
show running-config aclmgr	Displays ACLs in the running configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Default Settings

Table 15-1 lists the default settings for DAI parameters.

Table 15-1 Default DAI Parameters

Parameters	Default
DAI	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.

Table 15-1 Default DAI Parameters (continued)

Parameters	Default
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.

Additional References

For additional information related to implementing DAI, see the following sections:

- Related Documents, page 15-27
- Standards, page 15-27

Related Documents

Related Topic	Document Title
DHCP snooping	Information About DHCP Snooping, page 14-1
DAI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

Standards

Standards	Title
RFC-826	*An Ethernet Address Resolution Protocol* (http://tools.ietf.org/html/rfc826)

CHAPTER 16

Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on NX-OS devices.

This chapter includes the following sections:

Information 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 attacks that rely on 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 in 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 forward the packet only if the MAC address of the packet is 00:02:B3:3F:3B:99.

Virtualization Support

The following information applies to IP Source Guard used in Virtual Device Contexts (VDCs):

- IP-MAC address bindings are unique per VDC. Bindings in one VDC do not affect IP Source Guard in other VDCs.
- NX-OS does not limit binding database size on a per-VDC basis.

Licensing Requirements for IP Source Guard

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	IP Source Guard requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for IP Source Guard

IP Source Guard has the following prerequisites:

- You should be familiar with DHCP snooping before you configure IP Source Guard.
- DHCP snooping is enabled (see the "Configuring DHCP Snooping" section on page 14-6).

Guidelines and Limitations

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.

Configuring IP Source Guard

This section includes the following topics:

Enabling or Disabling IP Source Guard on a Layer 2 Interface

You can enable or disable IP Source Guard on a Layer 2 interface.

BEFORE YOU BEGIN

By default, IP Source Guard is disabled on all interfaces.

Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Snooping Feature" section on page 14-7.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. [**no**] **ip verify source dhcp-snooping-vlan**
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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** option disables IP Source Guard on the interface.
Step 4	**show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	(Optional) Displays the running configuration for DHCP snooping, including the IP Source Guard configuration.

	Command	Purpose
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) 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 a device.

BEFORE YOU BEGIN

By default, there are no static IP source entries on a device.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface ethernet** *slot/port*
3. **show ip dhcp snooping binding [interface ethernet** *slot/port*]
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	[**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface ethernet** *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, or if you use the **no** option, removes a static IP source entry.
Step 3	**show ip dhcp snooping binding** [**interface ethernet** *slot/port*] **Example:** `switch(config)# show ip dhcp snooping binding interface ethernet 2/3`	(Optional) Displays IP-MAC address bindings for the interface specified, including static IP source entries. Static entries appear with the term "static" in the Type column.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying the IP Source Guard Configuration

To display IP Source Guard configuration information, use one of the following commands:

Command	Purpose
show running-config dhcp	Displays DHCP snooping configuration, including the IP Source Guard configuration.
show ip dhcp snooping binding	Displays IP-MAC address bindings, including the static IP source entries.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying IP Source Guard Bindings

Use the **show ip verify source** command to display IP-MAC address bindings.

Example Configuration for IP Source Guard

The following example shows how to create a static IP source entry and then how to 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
```

Default Settings

Table 16-1 lists the default settings for IP Source Guard parameters.

Table 16-1 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.

Additional References

For additional information related to implementing IP Source Guard, see the following sections:

- Related Documents, page 16-6
- Standards, page 16-6

Related Documents

Related Topic	Document Title
Information About DHCP Snooping, page 14-1	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0*
IP Source Guard commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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.	—

CHAPTER 17

Configuring Keychain Management

This chapter describes how to configure keychain management on an NX-OS device.

This chapter includes the following sections:

Information About Keychain Management

This section includes the following topics:

Keychains and Keychain Management

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

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication. For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*.

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 key exchange with another device.
- Send lifetime—The time interval within which the device sends the key during 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.

Virtualization Support

The following information applies to keychains used in Virtual Device Contexts (VDCs):

- Keychains are unique per VDC. You cannot use a keychain that you created in one VDC in a different VDC.
- Because keychains are not shared by VDCs, you can reuse keychain names in different VDCs.
- The device does not limit keychains on a per-VDC basis.

Licensing Requirements for Keychain Management

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Keychain management requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Prerequisites for Keychain Management

Keychain management has no prerequisites.

Guidelines and Limitations

Keychain management has the following configuration guideline and limitation:

- Changing the system clock impacts the when keys are active.

Configuring Keychain Management

This section includes the following topics:

- Creating a Keychain, page 17-3
- Removing a Keychain, page 17-4
- Configuring a Key, page 17-5
- Configuring Text for a Key, page 17-6
- Configuring Accept and Send Lifetimes for a Key, page 17-7

Creating a Keychain

You can create a keychain on the device.

BEFORE YOU BEGIN

A new keychain contains no keys. For information about adding a key, see the "Configuring a Key" section on page 17-5.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **key chain** *name*
3. **show key chain** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**key chain** *name*
	
	Example:
	`switch(config)# key chain glbp-keys`
	`switch(config-keychain)#`
		Creates the keychain and enters keychain configuration mode.

Step 3	**show key chain** *name* **Example:** `switch(config-keychain)# show key chain glbp-keys`	(Optional) Displays the keychain configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config-keychain)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **no key chain** *name*
3. **show key chain** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no key chain** *name* **Example:** `switch(config)# no key chain glbp-keys`	Removes the keychain and any keys that the keychain contains.
Step 3	**show key chain** *name* **Example:** `switch(config-keychain)# show key chain glbp-keys`	(Optional) Confirms that the keychain no longer exists in running configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config-keychain)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Key

You can configure a key for a keychain.

A new key contains no text (shared secret). For information about adding text to a key, see the "Configuring Text for a Key" section on page 17-6.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

The default accept and send lifetimes for a new key are infinite. For more information, see the "Configuring Accept and Send Lifetimes for a Key" section on page 17-7.

SUMMARY STEPS

1. **config t**
2. **key chain** *name*
3. **key** *key-ID*
4. **show key chain** *name*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**key chain** *name*
	
	Example:
	`switch(config)# key chain glbp-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	**show key chain** *name* **Example:** `switch(config-keychain-key)# show key chain glbp-keys`	(Optional) Shows the keychain configuration, including the key configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-keychain)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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.

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.

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. For more information, see the “Configuring Accept and Send Lifetimes for a Key” section on page 17-7.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **key chain** *name*
3. **key** *key-ID*
4. **key-string** [*encryption-type*] *text-string*

5. **show key chain** *name* [**mode decrypt**]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**key chain** *name*
	
	Example:
	`switch(config)# key chain glbp-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 Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another NX-OS device.
Step 5	**show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain glbp-keys`	(Optional) Shows the keychain configuration, including the key text configuration. The **mode decrypt** option, which can be used by a device administrator only, displays the keys in cleartext.
Step 6	**copy running-config startup-config** **Example:** `switch(config-keychain-key)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Accept and Send Lifetimes for a Key

You can configure the accept lifetime and send lifetime for a key.

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.

BEFORE YOU BEGIN

By default, accept and send lifetimes for a key are infinite, which means that the key is always valid.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **key chain** *name*
3. **key** *key-ID*
4. **accept-lifetime** [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]
   send-lifetime [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]
5. **show key chain** *name* [**mode decrypt**]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	**key chain** *name* **Example:** switch(config)# key chain glbp-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.

	Command	Purpose
Step 4	**accept-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*] **Example:** `switch(config-keychain-key)# accept-lifetime 00:00:00 Jun 13 2008 23:59:59 Sep 12 2008`	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. 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.
	send-lifetime [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*] **Example:** `switch(config-keychain-key)# send-lifetime 00:00:00 Jun 13 2008 23:59:59 Aug 12 2008`	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 5	**show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain glbp-keys`	(Optional) Shows the keychain configuration, including the key text configuration. The **mode decrypt** option, which can be used by a device administrator only, displays the keys in cleartext.
Step 6	**copy running-config startup-config** **Example:** `switch(config-keychain-key)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Determining Active Key Lifetimes

To determine which keys within a keychain have active accept or send lifetimes, use the following command:

Command	Purpose
show key chain	Displays the keychains configured on the device.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Verifying the Keychain Management Configuration

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

Command	Purpose
show key chain	Displays the keychains configured on the device.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Example Configuration for Keychain Management

The following example shows how to configure a keychain named glbp 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 glbp-keys
  key 0
    key-string 7 zqdest
    accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
    send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
  key 1
    key-string 7 uaeqdyito
    accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
    send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008
  key 2
    key-string 7 eekgsdyd
    accept-lifetime 00:00:00 Nov 12 2008 23:59:59 Mar 12 2009
    send-lifetime 00:00:00 Dec 12 2008 23:59:59 Feb 12 2009
```

Where to Go Next

For information about routing features that use keychains, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*

Default Settings

Table 17-1 lists the default settings for keychain management parameters.

Table 17-1 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.

Additional References

For additional information related to implementing keychain management, see the following sections:

- Related Documents, page 17-11
- Standards, page 17-11

Related Documents

Related Topic	Document Title
Gateway Load Balancing Protocol	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*
Border Gateway Protocol	*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0*
Keychain management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

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.	—

CHAPTER 18

Configuring Traffic Storm Control

This chapter describes how to configure traffic storm control on the NX-OS device.

This chapter includes the following sections:

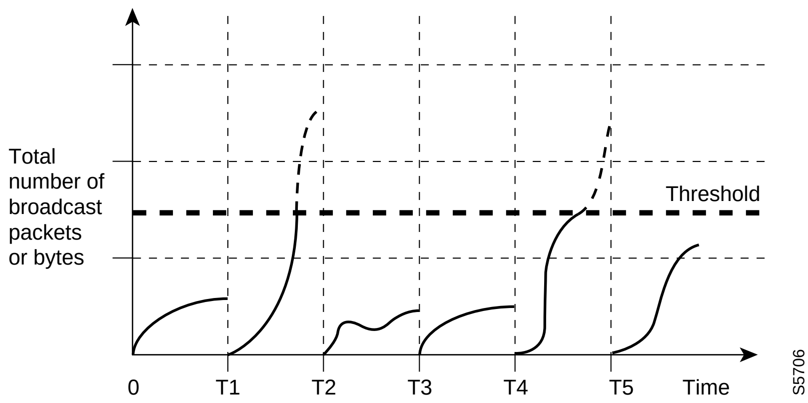
Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Layer 2 ports by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a 1-second interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

Figure 18-1 shows the broadcast traffic patterns on a Layer 2 interface over a given interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 18-1 Broadcast Suppression



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Cisco NX-OS device is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 1-second interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 1-second interval can affect the behavior of traffic storm control.

The following are examples of traffic storm control behavior:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 1-second interval, traffic storm control drops all broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within the 1-second interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 1-second interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 1-second interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

By default, the NX-OS software takes no corrective action when the traffic exceeds the configured level. However, you can configure an Embedded Event Management (EEM) action to error-disable an interface if the traffic does not subside (drop below threshold) within a certain time period. For information on configuring EEM, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0*.

Virtualization Support For Traffic Storm Control

Traffic storm control configuration and operation are local to the virtual device context (VDC).

For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

Licensing Requirements for Traffic Storm Control

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Traffic storm control requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Guidelines and Limitations

When configuring the traffic storm control level, note the following guidelines and limitations:

- You can configure traffic storm control on a port-channel interface.
- Do not configure traffic storm control on interfaces that are members of a port-channel interface. Configuring traffic storm control on interfaces that are configured as members of a port channel puts the ports into a suspended state.
- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.

Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

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

Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.

Note Traffic storm control uses a 1-second interval that can affect the behavior of traffic storm control.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
3. **storm-control** {**broadcast** | **multicast** | **unicast**} **level** *percentage*[**.***fraction*]
4. **exit**
5. **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t 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	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** *percentage*[**.***fraction*] **Example:** switch(config-if)# storm-control unicast level 40	Configures traffic storm control for traffic on the interface. The default state is disabled.
Step 4	**exit** **Example:** switch(config-if)# exit switch(config)#	Exits interface configuration mode.
Step 5	**show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*} **Example:** switch(config)# show running-config interface ethernet 1/1	(Optional) Displays the traffic storm control configuration.
Step 6	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Verifying Traffic Storm Control Configuration

To display traffic storm control configuration information, perform one of the following tasks:

Command	Purpose
show interface [**ethernet** *slot*/*port* | **port-channel** *number*] **counters storm-control**	Displays the traffic storm control configuration for the interfaces.
show running-config interface	Displays the traffic storm control configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying Traffic Storm Control Counters

You can display the counters the NX-OS device maintains for traffic storm control activity.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show interface** [**ethernet** *slot*/*port* | **port-channel** *number*] **counters storm-control**

DETAILED STEPS

	Command	Purpose
Step 1	`switch#` **`show interface`** `[`**`ethernet`** *`slot`*`/`*`port`* `|` **`port-channel`** *`number`*`]` **`counters storm-control`** **`Example:`** `switch# show interface counters storm-control`	Displays the traffic storm control counters.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Traffic Storm Control Example Configuration

The following example shows how to configure traffic storm control:

```
interface Ethernet1/1
storm-control broadcast level 40
storm-control multicast level 40
storm-control unicast level 40
```

Default Settings

Table 18-1 lists the default settings for traffic storm control parameters.

Table 18-1 Default Traffic Storm Control Parameters

Parameters	Default
Traffic storm control	Disabled.
Threshold percentage	100.

Additional References

For additional information related to implementing traffic storm control, see the following sections:

- Related Documents, page 18-6

Related Documents

Related Topic	Document Title
NX-OS Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

CHAPTER 19

Configuring Unicast RPF

This chapter describes how to configure Unicast Reverse Path Forwarding (Unicast RPF) on NX-OS devices.

This chapter includes the following sections:

Information About Unicast RPF

The Unicast RPF feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP 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 IP 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 device 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 device at the upstream end of a connection.

Unicast RPF verifies that any packet received at a device 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.

This section includes the following topics:

- Unicast RPF Process, page 19-2
- Per-Interface Statistics, page 19-3

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:

Step 1 Checks the input ACLs on the inbound interface.

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

Step 3 Conducts a FIB lookup for packet forwarding.

Step 4 Checks the output ACLs on the outbound interface.

Step 5 Forwards the packet.

Per-Interface Statistics

Each time a that the Cisco NX-OS software drops or forwards a packet at an interface, that information is counted: globally on the device and at each interface where you have applied Unicast RPF. Global statistics on dropped packets provide information about potential attacks on the network; however, these global statistics do not help to specify which interface is the source of the attack.

Per-interface statistics allow you to track two types of information about malformed packets:

- Unicast RPF drops
- Unicast RPF suppressed drops

The statistics on the number of packets that Unicast RPF drops help you to identify the interface that is the entry point of the attack. The Unicast RPF drop count tracks the number of drops at the interface.

The Unicast RPF suppressed drop count tracks the number of packets that failed the Unicast RPF check but were forwarded because of the permit permission set up in the ACL. Using the drop count and suppressed drop count statistics allow you to help isolate the attack at a specific interface.

Tip You can use ACL logging information to further identify the address or addresses that are being dropped by Unicast RPF.

Virtualization Support

Unicast RPF configuration and operation is local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Guidelines and Limitations

Unicast RPF has the following configuration guidelines and limitations:

- You must apply Unicast RPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream that you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF 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 Unicast

RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.

- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure Unicast RPF at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use Unicast RPF 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 Unicast RPF 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 Unicast RPF only where there is natural or configured symmetry.
- Unicast RPF 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.

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

Configuring Unicast RPF

You can configure one the following Unicast RPF modes on an ingress interface:

Strict Unicast RPF mode—A strict mode check is successful when Unicast RFP 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.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*
3. **ip verify unicast source reachable-via** {**any** [**allow-default**] | **rx**}
4. **exit**

5. **show ip interface ethernet** *slot*/*port*
6. **show running-config interface ethernet** *slot*/*port*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

Step 3	**ip 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. 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. The **rx** keyword specifies strict Unicast RPF.

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

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

Verifying Unicast RPF Configuration

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

Command	Purpose
show ip interface ethernet *slot/port*	Displays the IP-related information for an interface.
show running-config interface ethernet *slot/port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IP 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.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference, Release 4.0*.

Unicast RPF Example Configuration

The following example shows how to configure loose Unicast RFP:

```
interface Ethernet2/30
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure strict Unicast RFP:

```
interface Ethernet2/30
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
```

Default Settings

Table 19-1 lists the default settings for Unicast RPF parameters.

Table 19-1 Default Unicast RPF Parameters

Parameters	Default
Unicast RPF	Disabled

Additional References

For additional information related to implementing Unicast RPF, see the following sections:

- Related Documents, page 19-7

Related Documents

Related Topic	Document Title
Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

CHAPTER 20

Configuring Control Plane Policing

This chapter describes how to configure control plane policing (CoPP) on the NX-OS device.

This chapter includes the following sections:

Information About CoPP

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

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. Attacks on the supervisor module can be of various types such as DoS that generates IP traffic streams to the control plane at a very high rate. These attacks force the control plane to spend a large amount of time in handling these packets and prevents the control plane from processing genuine traffic.

Examples of DoS attacks are as follows:

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

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

- High supervisor CPU utilization.
- Loss of line protocol keep-alive messages and routing protocol updates, which lead to route flaps and major network outages.
- Interactive sessions using the CLI become slow or completely unresponsive due to high CPU utilization.
- Resources, such as the memory and buffers, might be unavailable for legitimate IP data packets.
- Packet queues fill up, which can cause indiscriminate packet drops.

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

This section includes the following topics:

- Control Plane Protection, page 20-2
- Modular QoS Command-Line Interface, page 20-8
- CoPP and the Management Interface, page 20-8
- Virtualization Support, page 20-8

Control Plane Protection

To protect the control plane, the NX-OS device segregates different packets destined to the control plane into different classes. Once these classes are identified, the NX-OS device polices or marks down packets, which ensure that the supervisor module is not overwhelmed.

This section includes the following topics:

- Control Plane Packet Types, page 20-3
- Classification, page 20-3
- Rate Controlling Mechanisms, page 20-3
- Default Policing Policies, page 20-4

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, then the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.
- Redirected packets—Packets that are redirected to the supervisor module. Features like Dynamic Host Configuration Protocol (DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.
- Glean packets—If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the 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 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. The following parameters that can be used for classifying a packet:

- Source IP address
- Destination IP address
- Source MAC address
- Destination MAC address
- VLAN
- Source port
- Destination port
- Exception cause

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 traffics that conforms to, exceeds, 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.
- Peak information rate (PIR)—Rate above which data traffic is negatively affected.
- Committed burst (BC)—Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling.
- Extended burst (BE)—Size that a traffic burst can reach before all traffic exceeds the PIR.

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

For more information on policing parameters, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0*.

Default Policing Policies

When you bring up your NX-OS device for the first time, the NX-OS software installs the default copp-system-policy 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 (1 rate and two color)
- Moderate (2 rate and 3 color where the PIR is 25 percent higher than the CIR of the strict default policy)
- Lenient (2 rate and 3 color where the PIR is 50 percent higher than the CIR of the strict default policy)
- None (no control plane policy is applied)

If you do not select an option or choose not to execute the setup utility, the NX-OS software applies strict policing. You can change the CoPP policies as needed from the CLI. You can also remove the default copp-system-policy from the CLI.

The copp-system-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 NX-OS software on your device.

Caution Selecting the none option and not subsequently configuring CoPP protection can leave your NX-OS device vulnerable to DoS attacks.

In Cisco NX-OS Release 4.0(2) and later releases, you can reassign the CoPP default policy by entering the setup utility again using the **setup** command from the CLI prompt. Any changes you have made to the CoPP configuration are lost. For an example of using the setup utility, see the "Changing or Reapplying the Default CoPP Policy" section on page 20-20.

Note If you are using a CoPP default policy, we recommend that you reapply the CoPP default policy using the **setup** command after you upgrade to Cisco NX-OS Release 4.0(2) or later releases (see the "Changing or Reapplying the Default CoPP Policy" section on page 20-16).

This section includes the following topics:

- Default Classes, page 20-5
- Strict Default CoPP Policy, page 20-6
- Moderate Default CoPP Policy, page 20-6
- Lenient Default CoPP Policy, page 20-7

Default Classes

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

```
class-map type control-plane match-any copp-system-class-critical
    match access-group name copp-system-acl-igmp
    match access-group name copp-system-acl-msdp
    match access-group name copp-system-acl-bgp
    match access-group name copp-system-acl-eigrp
    match access-group name copp-system-acl-rip
    match access-group name copp-system-acl-ospf
    match access-group name copp-system-acl-pim
```

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

```
class-map type control-plane match-any copp-system-class-important
    match access-group name copp-system-acl-hsrp
    match access-group name copp-system-acl-vrrp
    match access-group name copp-system-acl-glbp
    match access-group name copp-system-acl-pim-reg
```

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

```
class-map type control-plane match-any copp-system-class-management
    match access-group name copp-system-acl-tacacs
    match access-group name copp-system-acl-radius
    match access-group name copp-system-acl-ntp
    match access-group name copp-system-acl-ftp
    match access-group name copp-system-acl-tftp
    match access-group name copp-system-acl-sftp
    match access-group name copp-system-acl-ssh
    match access-group name copp-system-acl-snmp
    match access-group name copp-system-acl-telnet
```

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

```
class-map type control-plane match-any copp-system-class-normal
    match protocol arp
```

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

```
class-map type control-plane match-any copp-system-class-redirect
    match redirect arp-inspect
    match redirect dhcp-snoop
```

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

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

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

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

```
    match exception ipv6 option
    match exception ipv6 icmp unreachable
```

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

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

Strict Default CoPP Policy

The strict default CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-policy

    class copp-system-class-critical
        police cir 40900 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-important
        police cir 1060 kbps bc 1000 ms conform transmit violate drop

    class copp-system-class-management
        police cir 10000 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-normal
        police cir 680 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-redirect
        police cir 280 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-monitoring
        police cir 100 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-exception
        police cir 360 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-undesirable
        police cir 32 kbps bc 250 ms conform drop violate drop

    class class-default
        police cir 100 kbps bc 250 ms conform transmit violate drop
```

Moderate Default CoPP Policy

The moderate default CoPP policy has the following configuration:

```
class copp-system-class-critical
    police cir 40900 kbps bc 250 ms pir 51200 kbps be 250 ms conform transmit exceed 
transmit violate drop

class copp-system-class-important
    police cir 1060 kbps bc 1000 ms pir 1325 kbps be 1000 ms conform transmit exceed 
transmit violate drop

class copp-system-class-management
    police cir 10000 kbps bc 250 ms pir 12500 kbps be 250 ms conform transmit exceed 
transmit violate drop

class copp-system-class-normal
    police cir 680 kbps bc 250 ms pir 850 kbps be 250 ms conform transmit exceed transmit 
violate drop
```

```
class copp-system-class-redirect
    police cir 280 kbps bc 250 ms pir 350 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-monitoring
    police cir 100 kbps bc 250 ms pir 125 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-exception
    police cir 360 kbps bc 250 ms pir 450 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-undesirable
    police cir 32 kbps bc 250 ms conform drop violate drop

class class-default
    police cir 100 kbps bc 250 ms pir 125 kbps be 250 ms conform transmit exceed transmit
violate drop
```

Lenient Default CoPP Policy

The lenient default CoPP policy has the following configuration:

```
class copp-system-class-critical
    police cir 40900 kbps bc 250 ms pir 61350 kbps be 250 ms conform transmit exceed
transmit violate drop

class copp-system-class-important
    police cir 1060 kbps bc 1000 ms pir 1500 kbps be 1000 ms conform transmit exceed
transmit violate drop

class copp-system-class-management
    police cir 10000 kbps bc 250 ms pir 15000 kbps be 250 ms conform transmit exceed
transmit violate drop

class copp-system-class-normal
    police cir 680 kbps bc 250 ms pir 1020 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-redirect
    police cir 280 kbps bc 250 ms pir 420 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-monitoring
    police cir 100 kbps bc 250 ms pir 150 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-exception
    police cir 360 kbps bc 250 ms pir 550 kbps be 250 ms conform transmit exceed transmit
violate drop

class copp-system-class-undesirable
    police cir 32 kbps bc 250 ms conform drop violate drop

class class-default
    police cir 100 kbps bc 250 ms pir 150 kbps be 250 ms conform transmit exceed transmit
violate drop
```

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.

The MQC structure consists of the following high-level steps:

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

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.

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 not pass through the in-band traffic hardware where CoPP is implemented. To limit traffic on the mgmt0 interface, use ACLs (see Chapter 10, "Configuring IP ACLs" and Chapter 11, "Configuring MAC ACLs").

Virtualization Support

You can configure CoPP only in the default virtual device context (VDC), but the CoPP configuration applies to all VDCs on the NX-OS device. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

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 Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Guidelines and Limitations

CoPP has the following configuration guidelines and limitations:

- You must use the setup utility to change or reapply the default copp-system-policy policy. You can access the setup utility using the **setup** command at the CLI.

Send document comments to nexus7k-docfeedback@cisco.com

- CoPP does not support non-IP classes except for the default non-IP class. You can use ACLs instead of non-IP classes to drop non-IP traffic, and use the default non-IP CoPP class to limit to non-IP traffic that reaches the supervisor module.
- You cannot enable logging in CoPP policy ACLs.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the switches. 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 (service-policy output CoPP cannot be applied 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.

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

Configuring CoPP

This section includes the following topics:

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.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

Ensure that you have configured the IP ACLs (see Chapter 10, "Configuring IP ACLs") or MAC ACLs (see Chapter 11, "Configuring MAC ACLs") if you want to use ACE hit counters in the class maps.

SUMMARY STEPS

1. **config t**
2. **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*

3. **match access-group name** *access-list-name*

   match exception {**ip** | **ipv6**} **icmp redirect**

   match exception {**ip** | **ipv6**} **icmp unreachable**

   match exception {**ip** | **ipv6**} **option**

   match protocol arp

   match redirect arp-inspect

   match redirect dhcp-snoop
4. **exit**
5. **show class-map type control-plane** [*class-map-name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
	
	Example:
	`switch(config)# class-may 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.

	Command	Purpose
Step 3	**match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name MyAccessList`	Specifies matching for an IP ACL. You can repeat this step to match more than one IP ACL. **Note** The permit and deny ACL keywords are ignored in the control plane policing matching.
	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.
	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.
	match exception {**ip** | **ipv6**} **option** **Example:** `switch(config-cmap)# match exception ip option`	Specifies matching for IPv4 or IPv6 option exception packets.
	match protocol arp **Example:** `switch(config-cmap)# match protocol arp`	Specifies matching for IP Address Resolution Protocol (ARP) packets.
	match redirect arp-inspect **Example:** `switch(config-cmap)# match redirect arp-inspect`	Specifies matching for ARP inspection redirected packets.
	match redirect dhcp-snoop **Example:** `switch(config-cmap)# match redirect dhcp-snoop`	Specifies matching for Dynamic Host Configuration Protocol (DHCP) snooping redirected packets.

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

Step 5	**show class-map type control-plane** [*class-map-name*] **Example:** `switch(config)# show class-map type control-plane`	(Optional) Displays the control plane class map configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which include policing parameters. If you do not configure a policer for a class, then the default policer conform action is drop. Glean packets are policed using the default-class. The NX-OS software supports 1-rate 2-color and 2-rate 3-color policing.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

Ensure that you have configured a control plane class map (see the "Configuring a Control Plane Class Map" section on page 20-9).

SUMMARY STEPS

1. **config t**
2. **policy-map type control-plane** *policy-map-name*
3. **class** {*class-map-name* [**insert-before** *class-map-name*] | **class-default**}
4. **police** [**cir**] *cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]

   police [**cir**] *cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] [**bc**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]

   police [**cir**] *cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]
   conform {**drop** | **set-cos-transmit** *cos-value* | **set-dscp-transmit** *dscp-value* | **set-prec-transmit** *prec-value* | **transmit**} [**exceed** {**drop** | **set dscp dscp table cir-markdown-map** | **transmit**}] [**violate** {**drop** | **set dscp dscp table pir-markdown-map** | **transmit**}]

   police [**cir**] *cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]
   pir *pir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] [[**be**] *extended-burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]]
5. (Optional) **set cos** [**inner**] *cos-value*
6. (Optional) **set dscp** [**tunnel**] {*dscp-value* | **af11** | **af12** | **af13** | **af21** | **af22** | **af23** | **af31** | **af32** | **af33** | **af41** | **af42** | **af43** | **cs1** | **cs2** | **cs3** | **cs4** | **cs5** | **cs6** | **cs7** | **sf** | **default**}
7. (Optional) **set precedence** [**tunnel**] *prec-value*
8. **exit**
9. **exit**
10. **show policy-map type control-plane** [**expand**] [**name** *policy-map-name*]
11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

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

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

	Command	Purpose
Step 4	**police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] | **percent** *percent*} **Example:** `switch(config-pmap-c)# police cir 52000`	Specifies the committed information rate (CIR). The rate range is from 0 to 80000000000. The default CIR unit is **bps**.
	police [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] | **percent** *percent*} [**bc**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000`	Specifies the CIR with the committed burst (BC). The CIR range is from 0 to 80000000000 and the BC range is from 0 to 512000000. The default CIR unit is **bps** and the default BC size unit is **bytes**.
	police [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] | **percent** *percent*} **conform** {**drop** | **set-cos-transmit** *cos-value* | **set-dscp-transmit** *dscp-value* | **set-prec-transmit** *prec-value* | **transmit**} [**exceed** {**drop** | **set dscp dscp table cir-markdown-map** | **transmit**}] [**violate** {**drop** | **set dscp dscp table pir-markdown-map** | **transmit**}] **Example:** `switch(config-pmap-c)# police cir 52000 conform transmit exceed drop`	Specifies the CIR with the conform action. The CIR range is from 0 to 80000000000. The default rate unit is **bps**. The range for the *cos-value* and *prec-value* arguments is from 0 to 7. The range for the *dscp-value* argument is from 0 to 63. The options are as follows: • **drop**—Drops the packet. • **set-cos-transmit**—Sets the cost of service value. • **set-dscp-transmit**—Sets the differentiated services code point value. • **set-prec-transmit**—Sets the precedence value. • **transmit—**Transmits the packet. • **set dscp dscp table cir-markdown-map—**Sets the exceed action to the CIR markdown map. • **set dscp dscp table pir-markdown-map**—Sets the violate action to the PIR markdown map. **Note** You can specify the BC and conform action for the same CIR.
	police [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**] | **percent** *percent*} **pir** *pir-rate* [**bps** | **gbps** | **kbps**| **mbps**] [[**be**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]] **Example:** `switch(config-pmap-c)# police cir 52000 pir 78000 be 2000`	Specifies the CIR with the peak information rate (PIR). The CIR range is from 0 to 80000000000 and the PIR range is from 1 to 80000000000. You can optional set an extended burst (BE) size. The BE range is from 1 to 512000000. The default CIR unit is **bps**, the default PIR unit is **bps**, and the default BE size unit is **bytes**. **Note** You can specify the BC, conform action, and PIR for the same CIR.
Step 5	**set cos** [**inner**] *cos-value* **Example:** `switch(config-pmap-c)# set cos 1`	(Optional) Specifies the 802.1Q class of service (CoS) value. Use the **inner** keyword in a Q-in-Q environment. The range is from 0 to 7. The default value is 0.

Send document comments to nexus7k-docfeedback@cisco.com

	Command	Purpose
Step 6	**set dscp** [**tunnel**] {*dscp-value* | **af11** | **af12** | **af13** | **af21** | **af22** | **af23** | **af31** | **af32** | **af33** | **af41** | **af42** | **af43** | **cs1** | **cs2** | **cs3** | **cs4** | **cs5** | **cs6** | **cs7** | **ef** | **default**} **Example:** `switch(config-pmap-c)# set dscp 10`	(Optional) Specifies the differentiated services code point value in IPv4 and IPv6 packets. Use the **tunnel** keyword to set tunnel encapsulation. The range is from 0 to 63. The default value is 0.
Step 7	**set precedence** [**tunnel**] {*prec-value* | **critical** | **flash** | **flash-override** | **immediate** | **internet** | **network** | **priority** | **routine**} **Example:** `switch(config-pmap-c)# set precedence 2`	(Optional) Specifies the precedence value in IPv4 and IPv6 packets. Use the **tunnel** keyword to set tunnel encapsulation. The range is from 0 to 7. The default value is 0.

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

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

Step 10	**show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:** `switch(config)# show policy-map type control-plane`	(Optional) Displays the control plane policy map configuration.
Step 11	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

Ensure that you have configured a control plan policy map (see the "Configuring a Control Plane Policy Map" section on page 20-12).

SUMMARY STEPS

1. **config t**
2. **control-plane**
3. **service-policy input** *policy-map-name*

4. **exit**
5. **show running-config copp [all]**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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	**service-policy input** *policy-map-name* **Example:** `switch(config-cp)# service-policy input PolicyMapA`	Specify a policy map for the input traffic. Repeat this step if you have more than one policy map. Use the **no service-policy input** *policy-map-name* command to remove the policy from the control plane.

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

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

Changing or Reapplying the Default CoPP Policy

In Cisco NX-OS Release 4.0(2) and later releases, you can change to a different default CoPP policy using the setup utility. You can also reapply the same CoPP default policy. For an example of changing the default CoPP policy, see the "Changing or Reapplying the Default CoPP Policy" section on page 20-20.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **setup**

DETAILED STEPS

	Command	Purpose
Step 1	**`setup`** **Example:** `switch# setup`	Enters the setup utility.

Displaying the CoPP Configuration Status

In Cisco NX-OS Release 4.0(2) and later releases, you can display the CoPP feature configuration status information.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show copp status**

DETAILED STEPS

	Command	Purpose
Step 1	**`show copp status`** **Example:** `switch# show copp status`	Displays CoPP feature configuration status information.

For detailed information about the fields in the output from this command, see to the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Displaying the CoPP Statistics

You can display the CoPP statistics.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show policy-map interface control-plane**

DETAILED STEPS

	Command	Purpose
Step 1	**show policy-map interface control-plane** **Example:** `switch# show policy-map interface control-plane`	Displays control plane statistics.

For detailed information about the fields in the output from this command, see to the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Clearing the CoPP Statistics

You can clear the CoPP statistics.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show policy-map interface control-plane**
2. **clear copp statistics**

DETAILED STEPS

	Command	Purpose
Step 1	**show policy-map interface control-plane** **Example:** `switch# show policy-map interface control-plane`	(Optional) Displays control plane statistics.
Step 2	**clear copp statistics** **Example:** `switch# clear copp statistics`	Clears the CoPP statistics.

Verifying CoPP Configuration

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

Command	Purpose
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration.
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map configuration.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show startup-config copp	Displays the CoPP configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see to the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

CoPP Example Configurations

This section includes the following topics:

- CoPP Configuration Example, page 20-19
- Changing or Reapplying the Default CoPP Policy, page 20-20

CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-acl-arp
permit any any 0x0806

ip access-list copp-system-acl-tacas
permit udp any any eq 49

ip access-list copp-system-acl-gre
permit 47 any any

ip access-list copp-system-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-acl-icmp
permit icmp any any

class-map type control-plane match-any copp-system-class-critical
```

```
match access-group name copp-system-acl-igmp
match access-group name copp-system-acl-msdp
match access-group name copp-system-acl-arp

class-map type control-plane match-any copp-system-class-important
match access-group name copp-system-acl-tacas
match access-group name copp-system-acl-gre

class-map type control-plane match-any copp-system-class-normal
match access-group name copp-system-acl-icmp
match exception ip icmp redirect
match exception ip icmp unreachable
match exception ip option
match redirect arp-inspect
match redirect dhcp-snoop

policy-map type control-plane copp-system-policy
class copp-system-class-critical
police cir 2000 kbps bc 1500 bytes pir 3000 kbps be 1500 bytes conform transmit exceed
transmit violate drop

class copp-system-class-important
police cir 1000 kbps bc 1500 bytes pir 1500 kbps be 1500 bytes conform transmit exceed
transmit violate drop

class copp-system-class-normal
police cir 400 kbps bc 1500 bytes pir 600 kbps be 1500 bytes conform transmit exceed
transmit violate drop

class class-default
police cir 200 kbps bc 1500 bytes pir 300 kbps be 1500 bytes conform transmit exceed
transmit violate drop

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

Changing or Reapplying the Default CoPP Policy

This following example shows how to change or reapply the default CoPP policy using the setup utility:

```
switch# setup


         ---- Basic System Configuration Dialog VDC: 1 ----

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


  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/none) [strict]: strict

  Configure CMP processor on current sup (slot 6)? (yes/no) [y]: n

  Configure CMP processor on redundant sup (slot 5)? (yes/no) [y]: n

The following configuration will be applied:
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

Default Settings

Table 20-1 lists the default settings for CoPP parameters.

Table 20-1 Default CoPP Parameters

Parameters	Default
Default policy	Strict

Additional References

For additional information related to implementing CoPP, see the following sections:

- Related Documents, page 20-22
- Standards, page 20-22

Related Documents

Related Topic	Document Title
Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*
IP ACLs	Configuring IP ACLs
MAC ACLs	Configuring MAC ACLs

Standards

Standards	Title
RFC 2698	*A Two Rate Three Color Marker*

Feature History for CoPP

Table 20-2 lists the release history for this feature.

Table 20-2 Feature History for CoPP

Feature Name	Releases	Feature Information
Changing default policies	4.0(2)	You can use **setup** command to change the default CoPP policy.
Changes to the default policies	4.0(2)	The CoPP default policies have the following changes: • Added Secure Shell FTP (SFTP) to copp-system-class-management. • Added access-control lists (ACLs) to match the source ports for TACACS+, RADIUS, Network Time Protocol (NTP), FTP, TFTP, SFTP, Secure Shell (SSH), and Telnet. • Increased the policing bandwidth to 10 Mbps for the copp-system-class-management class. • Added an exception for IPv6 in the copp-system-class-exception class. • Added pim-reg in the copp-system-class-important class. • Enhanced the CoPP policies for HSRP and GLBP to improve scalability.

CHAPTER 21

Configuring Rate Limits

This chapter describes how to configure rate limits for egress traffic on NX-OS devices.

This chapter includes the following topics:

Information About Rate Limits

Rate limits can prevent redirected packets for egress exceptions from overwhelming the supervisor module on an NX-OS device. You can configure rate limits in packets per second for the following types of redirected packets:

- Access list logging packets
- Data and control packets copied to the supervisor module
- Layer 2 storm control packets
- Layer 2 port security packets
- Layer 3 glean packets
- Layer 3 maximum transmission unit (MTU) check failure packets
- Layer 3 multicast directly connected packets
- Layer 3 multicast local group packets

- Layer 3 multicast Reverse Path Forwarding (RPF) leak packets
- Layer 3 Time-to-Live (TTL) check failure packets
- Receive packets

You can also configure rate limits for Layer 3 control packets.

Virtualization Support

You can configure rate limits only in the default virtual device context (VDC), but the rate limits configuration applies to all VDCs on the NX-OS device. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*.

Licensing Requirements for Rate Limits

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	Rate limits require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

Guidelines and Limitations

Rate limits has the following configuration guidelines and limitations:

- You can set rate limits only for supervisor-bound egress exception and egress redirected traffic. Use control plane policing (CoPP) for other types of traffic (see Chapter 20, “Configuring Control Plane Policing”).

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

Configuring Rate Limits

You can set rate limits on egress traffic.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **platform rate-limit access-log-list** *packets*
   platform rate-limit copy *packets*
   platform rate-limit layer-2 port-security *packets*
   platform rate-limit layer-2 storm-control *packets*
   platform rate-limit layer-3 control *packets*
   platform rate-limit layer-3 glean *packets*
   platform rate-limit layer-3 mtu *packets*
   platform rate-limit layer-3 multicast {**directly-connected** | **local-groups** | **rpf-leak**} *packets*
   platform rate-limit layer-3 ttl *packets*
   platform rate-limit receive *packets*
3. **exit**
4. **show hardware rate-limit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

	Command	Purpose
Step 2	**platform rate-limit access-list-log** *packets* **Example:** `switch(config)# platform rate-limit 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 from1 to 33554431. The default rate is 100.
	platform rate-limit copy *packets* **Example:** `switch(config)# platform rate-limit copy 40000`	Configures rate limits in packets per second for data and control packets copied to the supervisor module. The range is from1 to 33554431. The default rate is 30000.
	platform rate-limit layer-2 port-security *packets* **Example:** `switch(config)# platform rate-limit control 100`	Configures rate limits in packets per second for port security packets. The range is from1 to 33554431. The default is disabled.
	platform rate-limit layer-2 storm-control *packets* **Example:** `switch(config)# platform rate-limit control 100`	Configures rate limits in packets per second for storm control packets. The range is from1 to 33554431. The default is disabled.
	platform rate-limit layer-3 control *packets* **Example:** `switch(config)# platform rate-limit control 20000`	Configures rate limits in packets per second for Layer-3 control packets. The range is from1 to 33554431. The default rate is 10000.
	platform rate-limit layer-3 glean *packets* **Example:** `switch(config)# platform rate-limit layer-3 glean 200`	Configures rate limits in packets per second for Layer-3 glean packets. The range is from1 to 33554431. The default rate is 100.
	platform rate-limit layer-3 mtu *packets* **Example:** `switch(config)# platform rate-limit layer-3 mtu 1000`	Configures rate limits in packets per second for Layer-3 MTU failure redirected packets. The range is from1 to 33554431. The default rate is 500.
	platform rate-limit layer-3 multicast {**directly-connected** | **local-groups** | **rpf-leak**} *packets* **Example:** `switch(config)# platform rate-limit layer-3 multicast local-groups 20000`	Configures rate limits in packets per second for Layer-3 multicast directly connected, local groups, or RPF leak redirected packets in packets per second. The range is from1 to 33554431. The default rate is 10000 for directly connected packets, 10000 for local groups packets, and 500 for RPF leak packets.
	platform rate-limit layer-3 ttl *packets* **Example:** `switch(config)# platform rate-limit layer-3 ttl 1000`	Configures rate limits in packets per second for Layer-3 failed Time-to-Live redirected packets. The range is from1 to 33554431. The default rate is 500.
	platform rate-limit receive *packets* **Example:** `switch(config)# platform rate-limit receive 40000`	Configures rate limits in packets per second for packets redirected to the supervisor module. The range is from1 to 33554431. The default rate is 30000.

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

	Command	Purpose
Step 4	**show hardware rate-limit** [**access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control**} | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**] **Example:** `switch# show running-config	include rate-limit`	(Optional) Displays the rate limit configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Displaying the Rate Limit Statistics

You can display the rate limit statistics.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show hardware rate-limit [access-list-log | copy | layer-2 storm-control | layer-3 {control | glean | mtu | multicast** {**directly-connected | local-groups | rpf-leak**} **| ttl} | receive]**

DETAILED STEPS

	Command	Purpose
Step 1	**show hardware rate-limit** [**access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control**} | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**] **Example:** `switch# show hardware rate-limit layer-3 glean`	Displays the rate limit statistics.

For detailed information about the fields in the output from this command, see to the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Clearing the Rate Limit Statistics

You can clear the rate limit statistics.

BEFORE YOU BEGIN

Ensure that you are in the default VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **show hardware rate-limit** [**access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control**}| **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**]
2. **clear hardware rate-limit** {**all** | **access-list-log** | **copy** | **layer-2 storm-control** | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**}

DETAILED STEPS

	Command	Purpose
Step 1	**show hardware rate-limit** [**access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control**} | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**] **Example:** `switch# show hardware rate-limit layer-3 glean`	(Optional) Displays the rate limit statistics.
Step 2	**clear hardware rate-limiter** {**all** | **access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control**} | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl**} | **receive**} **Example:** `switch# clear hardware rate-limiter`	Clears the rate limit statistics.

Verifying the Rate Limits Configuration

To display the rate limits configuration information, perform the following task:

Command	Purpose
1. **show hardware rate-limit** [**access-list-log** | **copy** | **layer-2** {**port-security** | **storm-control** | **layer-3** {**control** | **glean** | **mtu** | **multicast** {**directly-connected** | **local-groups** | **rpf-leak**} | **ttl}** | **receive**]	Displays the rate limit configuration.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*.

Rate Limits Example Configuration

The following example shows how to configure rate limits:

```
platform rate-limit layer-3 control 20000
platform rate-limit copy 40000
```

Default Settings

Table 21-1 lists the default settings for rate limits parameters.

Table 21-1 Default Rate Limits Parameters

Parameters	Default
Access-list-log packets rate limit	100 packets per second
Copy packets rate limit	30,000 packets per second
Layer 2 port-security packet rate limit	Disabled
Layer 2 storm-control packets rate limit	Disabled
Layer 3 control packets rate limit	10,000 packets per second
Layer 3 glean packets rate limit	100 packets per second
Layer 3 MTU packets rate limit	500 packets per second
Layer 3 multicast directly-connected packets rate limit	10,000 packets per second
Layer 3 multicast local-groups packets rate limit	10,000 packets per second
Layer 3 multicast rpf-leak packets rate limit	500 packets per second
Receive packets rate limit	30,000 packets per second

Additional References

For additional information related to implementing rate limits, see the following sections:

- Related Documents, page 21-8

Related Documents

Related Topic	Document Title
Licensing	*Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0*

Feature History for Rate Limits

Table 21-2 lists the release history for this feature.

Table 21-2 Feature History for IP ACLs

Feature Name	Releases	Feature Information
Port security packet rate limiting	4.0(3)	Rate limiting for port security packet was added to the **platform rate-limit**, **clear hardware rate-limit**, and **show hardware rate-limit** commands.
Rate limits	4.0(1)	This feature was introduced.

INDEX

Numerics

A

B

D

E

M

N

O

P

R

S

T