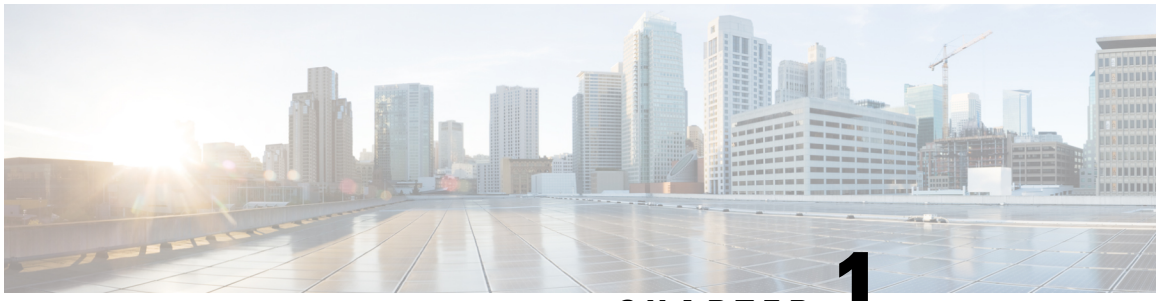
Cisco Nexus 1000V for VMware vSphere Security Configuration Guide, Release 5.x

First Published: 2014-08-22

Last Modified: 2019-05-20

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

CONTENTS



CHAPTER 1

New and Changed Information

This chapter lists new and changed content in this document by software release.

New and Changed Information for Security Configuration

This section lists new and changed content in this document by software release.

To find additional information about new features or command changes, see the *Cisco Nexus 1000V Release Notes* and *Cisco Nexus 1000V Command Reference*.

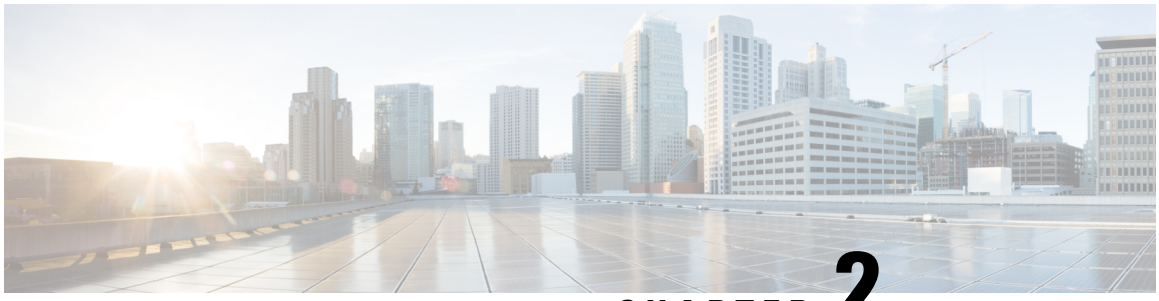
Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
802.1X Support	Support for 802.1X feature that defines a client-server-based access control and authentication protocol to restrict unauthorized clients from connecting to a LAN through publicly accessible ports.	5.2(1)SV3(4.1)	Information About 802.1X, on page 277
Cisco TrustSec Subnet-SGT Mapping	Support for binding SGT to all the host addresses of a specified subnet.	5.2(1)SV3(2.1)	Cisco TrustSec Subnet-SGT Mapping, on page 226
CTS SXPv3 Protocol Support	Support for Cisco TrustSec SXPv3 protocol.	5.2(1)SV3(2.1)	Cisco TrustSec With SXPv3, on page 225
Multi-IP per MAC support for IPSG	Multiple IP address attached to a MAC address for packet management.	5.2(1)SV3(2.1)	Information About IP Source Guard, on page 203

Feature	Description	Changed in Release	Where Documented
Cisco TrustSec SXP Peer Connection Modes	Cisco Nexus 1000V supports both speaker and listener modes for remote peer connections.	5.2(1)SV3(1.3)	Configuring Cisco TrustSec SXP Peer Connections, on page 242
Port Security	MAC Move Detection and Violation is no longer supported.	5.2(1)SV3(1.1)	Security Violations and Actions, on page 134
Layer 3 Security	Layer 3 Security (L3Sec) is a framework that secures the internal control plane communications (control and packet traffic) of the Cisco Nexus 1000V in a more robust way than in previous releases.	5.2(1)SV3(1.1)	Configuring Layer 3 Security, on page 275
Cisco TrustSec 2.0	This feature supports tagging of packets with the Cisco TrustSec command header and SGACL enforcement.	5.2(1)SV3(1.1)	Configuring Cisco TrustSec, on page 221
Traffic Storm Control	You can implement this feature to control broadcast, multicast, and unknown unicast traffic on ports and to control flooding.	5.2(1)SV3(1.1)	Configuring Traffic Storm Control, on page 269
SSH	SSH can support IPv6 addresses	5.2(1)SV3(1.1)	Configuring SSH, on page 83
Telnet	Telnet can support IPv6 addresses.	5.2(1)SV3(1.1)	Configuring Telnet, on page 95
IPACLs	You can configure IPv6 ACLs	5.2(1)SV3(1.1)	Configuring IP ACLs, on page 99
Cisco TrustSec	This feature was introduced.	4.2(1)SV2(1.1)	Configuring Cisco TrustSec, on page 221
Licensing Changes and advanced features	The following features are available as advanced features that require licenses: Cisco TrustSec, DHCP snooping, IP Source Guard, and Dynamic ARP Inspection.	4.2(1)SV2(1.1)	Configuring DHCP Snooping, on page 153,Configuring Dynamic ARP Inspection, on page 181,Configuring IP Source Guard, on page 203

Feature	Description	Changed in Release	Where Documented
DHCP Enhancements	You can enable source IP-based filtering on the Cisco Nexus 1000V switch.	4.2(1)SV2(1.1)	Configuring DHCP Snooping, on page 153
ACL Logging	You can log statistics for flows that match the ACL permit or deny conditions to monitor the flows.	4.2(1)SV1 (5.1)	Creating a MAC ACL, on page 122
UUFB	You can block unknown unicast packets from flooding the forwarding path.	4.2(1)SV1(4a)	Information About UUFB , on page 215
DHCP Snooping Relay Agent (Option 82)	You can configure DHCP to relay VSM MAC and port information in DHCP packets.	4.2(1)SV1(4)	Configuring DHCP Snooping, on page 153
DHCP Snooping binding table	You can clear DHCP snooping binding table entries for an interface.	4.2(1)SV1(4)	Configuring DHCP Snooping, on page 153
Enable DHCP	You can enable or disable DHCP globally by using the **feature DHCP** command.	4.2(1)SV1(4)	Configuring DHCP Snooping, on page 153
Enable SSH server	You can enable or disable the SSH server by using the **feature DHCP** command.	4.2(1)SV1(4)	Configuring SSH, on page 83
Enable Telnet server	You can enable or disable the Telnet server by using the **feature DHCP** command.	4.2(1)SV1(4)	Configuring Telnet, on page 95
Disable HTTP Server	You can disable the HTTP server for security purposes.	4.0(4)SV1(4)	Disabling the HTTP Server, on page 211
VSD	Virtual service domains (VSDs) allow you to classify and separate traffic for network services.	4.0(4)SV1(2)	Chapter 3, "Configuring VSD"
DHCP Snooping	The Dynamic Host Configuration Protocol (DHCP) snooping acts like a firewall between untrusted hosts and trusted DHCP servers.	4.0(4)SV1(2)	Configuring DHCP Snooping, on page 153
Dynamic ARP Inspection (DAI)	Dynamic ARP-inspection (DAI) provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address.	4.0(4)SV1(2)	Configuring Dynamic ARP Inspection, on page 181

Feature	Description	Changed in Release	Where Documented
IP Source Guard	IP Source Guard is a per-interface traffic permit filter for IP and MAC addresses.	4.0(4)SV1(2)	Configuring IP Source Guard, on page 203
Secure Login Enhancement	Support to configure login parameters.	5.2(1)SV3(4.1a)	Secure Login Enhancements, on page 40



CHAPTER 2

Overview

This chapter contains the following sections:

User Accounts

Access to the Cisco Nexus 1000V is accomplished by setting up user accounts that define the specific actions permitted by each user. You can create up to 256 user accounts. For each user account, you define a role, user name, password, and expiration date.

Virtual Service Domain

A virtual service domain (VSD) allows you to classify and separate traffic for network services, such as firewalls, traffic monitoring, and those in support of compliance goals.

Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) is an architectural framework for configuring a set of three independent, consistent, and modular security functions

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

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

RADIUS Security Protocol

AAA establishes communication between your network access server and your RADIUS security server. RADIUS is 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+ Security Protocol

AAA establishes communication between your network access server and your TACACS+ security server.

TACACS+ is 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 that usually runs on a UNIX or Windows NT workstation. TACACS+ provides separate and modular authentication, authorization, and accounting facilities.

SSH

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

The SSH client works with publicly and commercially available SSH servers.

Telnet

You can use the Telnet protocol to set up TCP/IP connections to a host. Telnet allows a person 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.

Access Control Lists

IP ACLs

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

MAC ACLs

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

Port Security

Port security allows you to configure Layer 2 interfaces permitting inbound traffic from a restricted and secured set of MAC addresses. Traffic from a secured MAC address is not allowed on another interface within the same VLAN. The number of MAC addresses that can be secured is configured per interface.

DHCP Snooping

DHCP snooping provides a mechanism to prevent a malicious host masquerading as a DHCP server from assigning IP addresses (and related configuration) to DHCP clients. In addition, DHCP snooping prevents certain denial of service attacks on the DHCP server.

DHCP snooping requires you to configure a trust setting for ports, which is used to differentiate between trusted and untrusted DHCP servers.

In addition, DHCP snooping learns IP addresses assigned by the DHCP server, so that other security features (for example, Dynamic ARP inspection and IP source guard) can function when DHCP is used to assign IP addresses to interfaces.

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) ensures that only valid ARP requests and responses are relayed by intercepting all ARP requests and responses on untrusted ports and verifying 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. When this feature is enabled, invalid ARP packets are dropped.

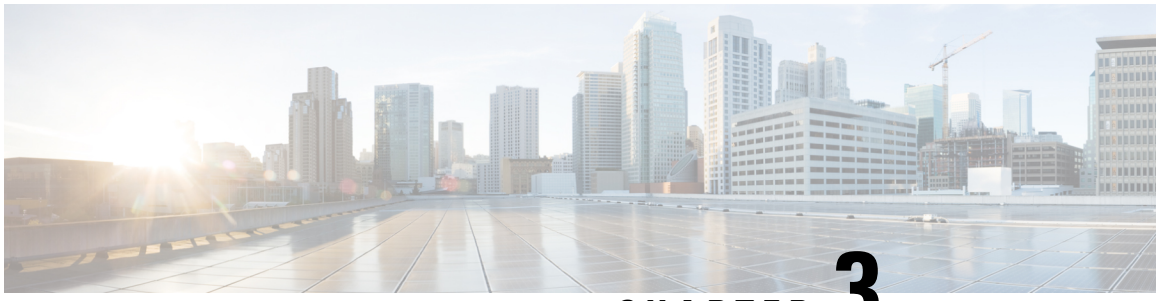
IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the packet IP address and MAC address match one of the following:

- The IP address and MAC address in the DHCP snooping binding
- The static IP source entries that you configure

Layer 3 Security

Layer 3 Security (L3Sec) is a framework that secures the internal control plane communications (control and packet traffic) of the Cisco Nexus 1000V in a more robust way than in previous releases. It operates only in Layer 3 Control mode.



CHAPTER 3

Managing User Accounts

This chapter contains the following sections:

Information About User Accounts

Access to the Cisco Nexus 1000V is accomplished by setting up user accounts that define the specific actions permitted by each user. You can create up to 256 user accounts. Each user account includes the following criteria:

- Role
- Username
- Password
- Expiration date

Role

A role is a collection of rules that define the specific actions that can be shared by a group of users. The following broadly defined roles, for example, can be assigned to user accounts. These roles are predefined in the Cisco Nexus 1000V and cannot be modified:

```
role: network-admin
  description: Predefined network admin role has access to all commands
  on the switch
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
```

```
  1       permit  read-write

role: network-operator
  description: Predefined network operator role has access to all read
  commands on the switch
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  1       permit  read
```

You can create an additional 64 roles that define access for users.

Each user account must be assigned at least one role and can be assigned up to 64 roles.

You can create roles that, by default, permit access to the following commands only. You must add rules to allow users to configure features.

- **show**
- **exit**
- **end**
- **configure terminal**

Username

A username identifies an individual user by a unique character string, such as daveGreen. Usernames are case sensitive and can consist of up to 28 alphanumeric characters. A username consisting of all numerals is not allowed. If an all-numeric username exists on an AAA server and is entered during login, the user is not logged in.

Password

A password is a case-sensitive character string that enables access by a specific user and helps prevent unauthorized access. You can add a user without a password, but they may not be able to access the device. Passwords should be strong so that they cannot be easily guessed for unauthorized access.

The following characters are not permitted in clear text passwords:

- dollar signs ($)
- spaces

The following special characters are not permitted at the beginning of the password:

- quotation marks (" or ')
- vertical bars (|)
- right angle brackets (>)

The following table lists the characteristics of strong passwords.

Table 2: Characteristics of Strong Passwords

Strong passwords have:	Strong passwords do not have:
At least eight characters	Consecutive characters, such as “abcd”
Uppercase letters	Repeating characters, such as “aaabbb”
Lowercase letters	Dictionary words
Numbers	Proper names
Special characters	

Some examples of strong passwords are as follows:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Check of Password Strength

The device checks password strength automatically by default. When you add a username and password, the strength of the password is evaluated. If it is a weak password, the following error message is displayed to notify you:

```
switch# config terminal
switch (config)# username daveGreen password davey

password is weak
Password should contain characters from at least three of the classes:
 lower case letters, upper case letters, digits, and special characters
```

Password strength checking can be disabled.

Expiration Date

By default, a user account does not expire. You can, however, explicitly configure an expiration date on which the account will be disabled.

Guidelines and Limitations for Creating User Accounts

- You can create up to 64 roles in addition to the two predefined user roles.
- You can create up to 256 rules in a user role.
- You can create up to 64 feature groups.
- You can add up to 256 users.
- You can assign a maximum of 64 user roles to a user account.

- If you have a user account that has the same name as a remote user account on an AAA server, the user roles for the local user account are applied to the remote user, not the user roles configured on the AAA server.

Guidelines for Creating User Accounts

- You can add up to 256 user accounts
- Changes to user accounts do not take effect until the user logs in and creates a new session.
- Do not use the following words in user accounts. These words are reserved for other purposes

adm	gdm	mtuser	rpcuser
bin	gopher	neews	shutdown
daemon	haltlp	nobody	sync
ftp	mail	nscd	sys
ftpuser	mailnull	operator	uucp
games	man	rpc	xfs

- You can add a user password as either clear text or encrypted.
  - Clear text passwords are encrypted before they are saved to the running configuration.
  - Encrypted passwords are saved to the running configuration without further encryption.
- A user account can have up to 64 roles, but must have at least one role.
- If you do not specify a password, the user might not be able to log in
- For information about using SSH public keys instead of passwords, see Configuring an OpenSSH Key, on page 86.

Default Settings for User Access

Parameters	Default
User account password	Undefined
User account expiration date	None
User account role	Network-operator
Interface policy	All interfaces are accessible
VLAN policy	All VLANs are accessible

Configuring User Access

Enabling the Check of Password Strength

You can enable the Cisco Nexus 1000V to check the strength of passwords to avoid creating weak passwords for user accounts.

Checking password strength is enabled by default. This procedure can be used to enable it again should it become disabled.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **password strength-check**	Enables password-strength checking. The default is enabled. You can disable the checking of password strength by using the **no** form of this command.
Step 3	(Optional) switch(config)# **show password strength-check**	Displays the configuration for checking password strength.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to check the strength of your password:

```
switch# configure terminal
switch(config)# password strength-check
switch(config)# show password strength-check
Password strength check enabled
switch(config)# copy running-config startup-config
```

Disabling the Check of Password Strength

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no password strength-check**	Disables password-strength checking. The default is enabled.
Step 3	(Optional) switch(config)# **show password strength-check**	Displays the configuration for checking password strength.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable the check of password strength:

```
switch# configure terminal
switch(config)# no password strength-check
switch(config)# show password strength-check
switch(config)# copy running-config startup-config
```

Creating a User Account

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **show role**	Displays the available roles that can be assigned to users.
Step 3	switch(config)# **username** *name* [**password** [**0** | **5**] *password*] [**expire date**] [**role** *role-name*]	Creates a user account. The arguments and keywords are as follows: • **username** *name*—A case-sensitive, alphanumeric character string of up to 28 characters in length. • **password** *password*—The default password is undefined. • **0**—(the default) Specifies that the password you are entering is in clear

	Command or Action	Purpose
		text. The Cisco Nexus 1000V encrypts the clear text password before saving it in the running configuration. In the example shown, the password 4Ty18Rnt is encrypted in your running configuration in password 5 format. • **5**—Specifies that the password you are entering is already in encrypted format. The Cisco Nexus 1000V does not encrypt the password before saving it in the running configuration. User passwords are not displayed in the configuration files. • **expire date**—YYYY-MM-DD. The default is no expiration date. • *role-name* **role**—You must assign at least one role. You can assign up to 64 roles. The default role is **network-operator**
Step 4	switch(config)# **show user-account** *username*	Displays the new user account configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a user account:

```
switch# configure terminal
switch(config)# show role
switch(config)# username NewUser password 4Ty18Rnt
switch(config)# show user-account NewUser
user: NewUser
     this user account has no expiry date
     roles:network-operator network-admin
switch# copy running-config startup-config
```

Creating a Role

Before you begin

- Log in to the CLI in EXEC mode.

- Know that you can configure up to 64 user roles.
- Know that you can configure up to up to 256 rules for each role.
- Know that you can assign a single role to more than one user.
- Know that the rule number specifies the order in which it is applied, in descending order. For example, if a role has three rules, rule 3 is applied first, rule 2 is applied next, and rule 1 is applied last.
- Know that 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 configure features.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role name** *role-name*	Names a user role and places you in role configuration mode for that role. The *role-name* is a case-sensitive, alphanumeric string of up to 16 characters.
Step 3	(Optional) switch(config-role)# **description** *description-string*	Configures the role description, which can include spaces.
Step 4	switch(config-role)# **rule** *number* {**deny**| **permit**} **command** *command-string* • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} Creates one rule to permit or deny all operations. • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name* Creates a rule for feature access. Use the **show role feature** command to display a list of available features. • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name* Creates a rule for feature group access. Use the **show role feature-group** command to display a list of feature groups. **Example:**	Creates a rule to permit or deny a specific command. The command you specify can contain spaces and regular expressions. For example, **interface ethernet *** permits or denies access to all Ethernet interfaces.

	Command or Action	Purpose
	This example configures a rule that denies access to the **clear users** command.	
Step 5	Repeat Step 4 to create all needed rules for the specified role.	
Step 6	(Optional) switch(config-role)# **show role**	Displays the user role configuration.
Step 7	(Optional) switch(config-role)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a role:

```
switch# configure terminal
switch(config)# role name UserA
switch(config-role)# description Prohibits use of clear commands
switch(config-role)# rule 1 deny command clear users
switch(config-role)# rule 2 deny read-write
switch(config-role)# rule 3 permit read feature eth-port-sec
switch(config-role)# rule 4 deny read-write feature-group eth-port-sec

switch# configure terminal
switch(config)# role name UserA
switch(config-role)# rule 3 permit read feature snmp
switch(config-role)# rule 2 permit read feature dot1x
switch(config-role)# rule 1 deny command clear *
```

Creating a Feature Group

You can create and configure a feature group. You can create up to 64 custom feature groups.

Before you begin

- Log in to the CLI in EXEC mode.
- Know that you can create up to 64 custom feature groups.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role feature-group name** *group-name*	Places you into the role feature group configuration mode for the named group. The *group-name* argument is case-sensitive, alphanumeric string of up to 32 characters in length.

	Command or Action	Purpose
Step 3	switch(config-role-featuregrp)# **show role feature**	Displays a list of available features for use in defining the feature group.
Step 4	switch(config-role-featuregrp)# **feature** *feature-name*	Adds a feature to the feature group. Repeat this step for all features to be added to the feature group.
Step 5	(Optional) switch(config-role-featuregrp)# **show role feature-group**	Displays the feature group configuration.
Step 6	(Optional) switch(config-role-featuregrp)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a feature group named GroupA:

```
switch# configure terminal
switch(config)# role feature-group name GroupA
switch(config-role-featuregrp)# show role feature
feature: aaa
feature: access-list
feature: cdp
feature: install
. . .
switch(config-role-featuregrp)# feature syslog
switch(config-role-featuregrp)# show role feature-group
feature group: GroupA
feature: syslog
feature: snmp
feature: ping
switch(config-role-featuregrp)# copy running-config startup-config
```

This example shows how to create a feature group named Security-features:

```
switch# configure terminal
switch(config)# role feature-group name Security-features
 switch(config-role-featuregrp)# feature radius
 switch(config-role-featuregrp)# feature tacacs
 switch(config-role-featuregrp)# feature dot1x
 switch(config-role-featuregrp)# feature aaa
 switch(config-role-featuregrp)# feature snmp
 switch(config-role-featuregrp)# feature acl
 switch(config-role-featuregrp)# feature access-list
```

Configuring Interface Access

By default, a role allows access to all interfaces. You modify a role that you have already created by denying access to all interfaces and then permitting access to selected interfaces.

Before you begin

- Log in to the CLI in EXEC mode
- You must have created one or more user roles. In this procedure, you are modifying a role that you have already created.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role name** *role-name*	Specifies a user role and enters role configuration mode for the named role.
Step 3	switch(config-role)# **interface policy deny**	Enters the interface configuration mode and denies all interface access for the role. Access to any interface must now be explicitly defined for this role by using the **permit interface** command
Step 4	switch(config-role-interface)# **permit interface** *interface-list*	Specifies the interface(s) that users assigned to this role can access. Repeat this command to specify all interface lists that users assigned to this role are permitted to access.
Step 5	(Optional) switch(config-role-interface)# **show role** *role-name*	Displays the role configuration.
Step 6	(Optional) switch(config-role-featuregrp)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure interface access:

```
switch# configure terminal
switch(config)# role name network-observer
switch(config-role)# interface policy deny
switch(config-role-interface)# permit interface ethernet 2/1-4
switch(config-role-interface)# show role name network-observer
  Role: network-observer
   Description: new role
switch(config-role-featuregrp)# copy running-config startup-config
```

Configuring VLAN Access

By default, access is allowed to all VLANs. In this procedure you are modifying a role that you have already created by denying access to all VLANs and then permitting access to selected VLANs.

Before you begin

- Log in to the CLI in EXEC mode.
- You must have already created one or more user roles. In this procedure, you are modifying a role that you have already created.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role name** *role-name*	Specifies a user role and enters role configuration mode.
Step 3	switch(config-role)# **vlan policy deny**	Enters the VLAN configuration mode and denies all VLAN access for the role. Access to any VLAN must now be explicitly defined for this role by using the **permit vlan** command.
Step 4	switch(config-role-vlan)# **permit vlan** *vlan-range*	Specifies the VLANs that users assigned to this role can access. Specify a VLAN range by using a dash. For example, 1-9 or 20-30. Repeat this command to specify all VLANs that users assigned to this role are permitted to access.
Step 5	(Optional) switch(config-role)# **show role** *role-name*	Displays the role configuration. The *role-name* argument is the name that you have assigned to the role you created.
Step 6	(Optional) switch(config-role)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure VLAN access:

```
switch# configure terminal
switch(config)# role name network-observer
switch(config-role)# vlan policy deny
switch(config-role-vlan)# permit vlan 1-4
switch(config-role)# show role name network-observer
  Role: network-observer
   Description: new role
  switch(config-role)# copy running-config startup-config
```

Verifying the User Access Configuration

Use one of the following commands to verify the configuration.

Command	Purpose
show role	Displays the available user roles and their rules.
show role feature	Displays a list of available features.
show role feature-group	Displays a list of available feature groups.
show startup-config security	Displays the user account configuration in the startup configuration.
show running-config security [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.
show user-account	Displays user account information.

Configuration Examples

Configuration Example for Creating a Feature Group

This example shows how to create a feature group:

```
switch# configure terminal
switch(config-role)# role feature-group name security-features
switch(config-role)# feature radius
switch(config-role)# feature tacacs
switch(config-role)# feature dot1x
switch(config-role)# feature aaa
switch(config-role)# feature snmp
switch(config-role)# feature acl
switch(config-role)# feature access-list
```

Configuration Example for Creating a Role

This example shows how to create a role:

```
switch# config terminal
switch(config)# role name UserA
switch(config-role)# rule 3 permit read feature snmp
switch(config-role)# rule 2 permit read feature dot1x
switch(config-role)# rule 1 deny command clear *
```

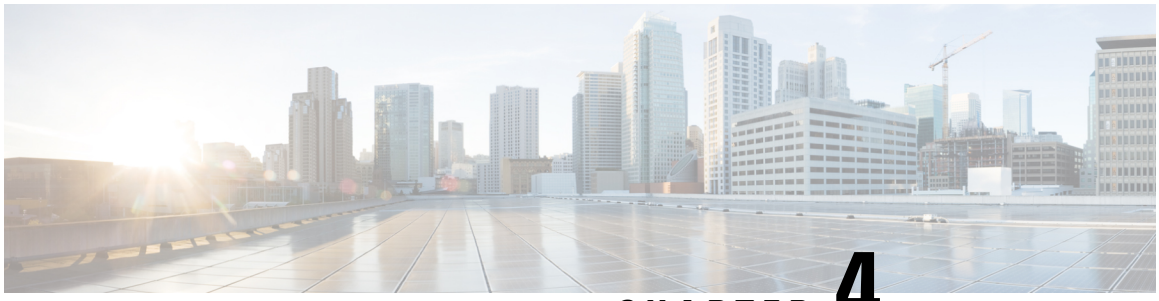
MIBs

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

Feature History for User Accounts

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
User Accounts	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 4

Configuring VSD

This chapter contains the following sections:

Information about Virtual Service Domains

A virtual service domain (VSD) allows you to classify and separate traffic for network services, such as firewalls, traffic monitoring, and those network services that are in support of compliance goals such as the Sarbanes Oxley Act.

Service Virtual Machine

A service virtual machine (SVM) provides the specialized service such as firewall, deep packet inspection (application aware networking), or monitoring. Each SVM has three virtual interfaces:

Interface	Description
Management	A regular interface that manages the SVM. This interface should have Layer 2 or Layer 3 connectivity, depending on its use.
Incoming	Guards the traffic coming into the VSD. Any packet coming into the VSD must go through this interface.
Outgoing	Guards the traffic going out of the VSD.. Any packet that originates in the VSD and goes out must go through the SVM and out through the outgoing interface.

There is no source MAC learning on these interfaces. Each SVM creates a secure VSD. Interfaces within the VSD are shielded by the SVM.

Port Profiles

A VSD is the collection of interfaces that are guarded by the SVM providing the security service. Any traffic coming into the VSD or going out of the VSD has to go through the SVM.

Traffic that both originates and terminates within the same VSD does not need to be routed through the SVM because it is considered to be safe.

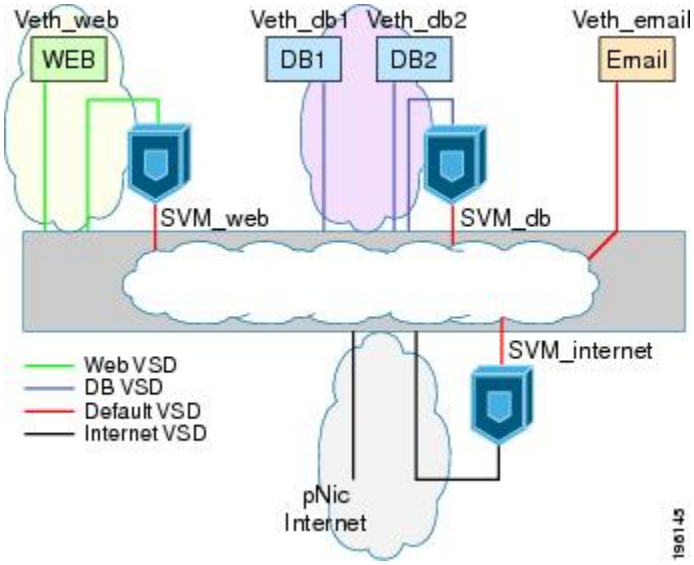
A VSD is formed by creating the following port profiles:

Port Profile	Description
Inside	Traffic originating from a VSD member goes into the service VM (SVM) through the inside port and comes out of the outside port before it is forwarded to its destination.
Outside	Traffic destined for a VSD member goes into the SVM through the outside port and comes out of the inside port before it is forwarded to its destination.
Member	Location for individual inside VMs.

The following diagram shows that a single VEM takes the place of vSwitches. The SVMs define the following VSDs in the diagram.

VSD	SVM (guard)	Inside Port Profile	Outside Port Profile	Member Port Profile(s)
DB VSD	SVM_db	SVM_db_inside	SVM_db_outside	vEth_db1 vEth_db2
Web VSD	SVM_web	SVM_web_inside	SVM_web_outside	vEth_web
Internet VSD	SVM_Internet	SVM_internet_inside	SVM_internet_outside	
Default		SVM VSD		vEth Email

Figure 1: Virtual Service Domain Example



Guidelines and Limitations

- To prevent traffic latency, VSD should only be used for securing traffic.
- Up to 6 VSDs can be configured per host and up to 64 on the VSM.
- Up to 214 interfaces per VSD are supported on a single host, and 2048 interfaces on the VSM.
- Vmotion is not supported for the SVM and should be disabled.
- To avoid network loops following a VSM reload or a network disruption, control and packet VLANs must be disabled in all port profiles of the Service VMs.
- If a port profile without a service port is configured on an SVM, it will flood the network with packets.
- When configuring a port profile on an SVMs, first bring the SVM down, This action prevents a port profile that is mistakenly configured without a service port from flooding the network with packets. The SVM can be returned to service after the configuration is complete and verified.
- VShield 4.1 does not support VSD. The VSD feature will not function as expected if used with VShield 4.1.

Default Settings

Table 3: Telnet Default Settings

Parameters	Default
service-port default-action	Forward

Parameters	Default
switchport trunk allowed vlan	All

Configuring VSD

Configuring an Inside or Outside VSD Port Profile

Use this procedure to configure the port profiles that define the connections going into and out of the SVM. While performing this procedure, keep in mind the following points:

- If you do not configure a service port, the SVM will come up as a regular VM and flood the network with packets.
- Selected VLAN filtering is not supported in this configuration. The default should be used instead, which allows all VLANs on the port.

Before you begin

Before beginning this procedure, be sure you:

- Are logged in to the CLI in EXEC mode.
- Have taken the SVM out of service to prevent any configuration errors from flooding the network. Once the configuration is complete and verified, you can bring the SVM back into service.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile***name*	Creates a port profile and places you into port profile configuration mode for the named port profile. The name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
Step 3	switch(config-port-profile)# **switchport mode trunk**	Designates that the interfaces are switch trunk ports.
Step 4	switch(config-port-profile)# **switchport trunk allowed vlan***vlanID*	Allows all VLANs on the port.
Step 5	switch(config-port-profile)# **virtual-service-domain** *name*	Adds a VSD name to this port profile.
Step 6	switch(config-port-profile)# **no shutdown**	Administratively enables all ports in the profile.

	Command or Action	Purpose
Step 7	switch(config-port-profile)# **vmware port-group** *pg-name*	Designates the port profile as a VMware port-group. The port profile is mapped to a VMware port group of the same name. When a vCenter Server connection is established, the port group created in Cisco Nexus 1000V is then distributed to the virtual switch on the vCenter Server. pg-name—Port group name. If you do not specify a pg-name, the port group name will be the same as the port profile name. If you want to map the port profile to a different port group name, use the pg-name option followed by the alternate name.

Step 8	switch(config-port-profile)# **service-port** { **inside** | **outside** } [**default-action** { **drop** | **forward** }]
	Example:
	`switch(config-port-profile)# service-port inside default-action forward`
	This example configures an inside VSD that forwards packets if the service port is down.
	Example:
	`switch(config-port-prof)# service-port outside default-action forward`
	This example configures an outside VSD that forwards packets if the service port is down.
		Configures the interface as either inside or outside and designates (default action) whether packets should be forwarded or dropped if the service port is down.
		This command has the following variables:
		• inside—Inside network
		• outside—Outside network
		• default-action — (Optional) Action to be taken if service port is down.
		• drop—drops packets
		• forward: forwards packets
		If you do not specify a default action, then the **forward** setting is used by default.
		Caution If you do not configure a service port, the SVM will come up as a regular VM, flooding the network with packets.

Step 9	switch(config-port-profile)# **state enabled**	Enables the VSD port profile. The configuration for this port profile is applied to the assigned ports, and the port group is created in the VMware vSwitch on the vCenter Server.
Step 10	(Optional) switch(config-port-profile)# **show virtual-service-domain name**	Displays the configuration for this VSD port profile. Use this to verify that the port profile was configured as expected. name—The name of the VSD.

	Command or Action	Purpose
Step 11	(Optional) switch(config-port-profile)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

```
switch# config terminal
switch(config)# port-profile webserver-inside
switch(config-port-profile)# switchport mode trunk
switch(config-port-profile)# switchport trunk allowed vlan all
switch(config-port-profile)# virtual-service-domain vsd1-webserver
switch(config-port-prof)# no shutdown
switch(config-port-prof)# vmware port-group webservers-inside-protected
switch(config-port-prof)# service-port inside default-action forward
switch(config-port-prof)# state enabled
switch(config-port-prof)# show virtual-service-domain vsd1-webserver
Default Action: forward

_____________________________
Interface        Type

_____________________________
Vethernet1       Member
Vethernet2       Member
Vethernet3       Member
Vethernet7       Inside
Vethernet8       Outside
switch(config-port-prof)# copy running-config startup-config
[########################################] 100%
```

Configuring a Member VSD Port Profile

Use this procedure to configure the VSD port profile where individual members reside.

Do not configure a member VSD port profile on an SVM. A member VSD port profile does not have a service port, and will flood the network with packets if configured on an SVM.

Before you begin

Before beginning this procedure, you must be logged in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Creates a port profile and places you in port profile configuration mode for the named port profile. The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.

	Command or Action	Purpose
Step 3	switch(config-port-profile)# **switchport access vlan** *vlanID*	Assigns a VLAN ID to the access port for this port profile. VLAN ID—The VLAN identification number. The range of valid values is 1 to 3967.
Step 4	switch(config-port-profile)# **virtual-service-domain** *name*	Created and names a VSD for this port profile
Step 5	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 6	switch(config-port-prof)# **state enabled**	Enables the VSD port profile. The configuration for this port profile is applied to the assigned ports, and the port group is created in the VMware vSwitch on the vCenter Server.
Step 7	(Optional) switch(config-port-prof)# **show virtual-service-domain** *name*	Displays the configuration for this VSD port profile. Use this to verify that the port-profile was configured as expected
Step 8	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

```
switch# configure terminal
switch(config)# port-profile vsd1-member
n1000v(config-port-profile)# switchport access vlan 315
n1000v(config-port-profile)# virtual-service-domain vsd1-webserver
n1000v(config-port-prof)# no shutdown
n1000v(config-port-prof)# state enabled
n1000v(config-port-prof)# show virtual-service-domain vsd1-webserver
Default Action: forward

_______________________________
Interface        Type

_______________________________
Vethernet1       Member
Vethernet2       Member
Vethernet3       Member
Vethernet6       Member
Vethernet7       Inside
Vethernet8       Outside
n1000v(config-port-prof)# copy running-config startup-config
[########################################] 100%


n1000v# config t
n1000v(config)# port-profile vsd1_member
n1000v(config-port-profile)# vmware port-group
n1000v(config-port-profile)# switchport access vlan 315
n1000v(config-port-profile)# virtual-service-domain vsd1
n1000v(config-port-profile)# no shutdown
  state enabled
n1000v(config-port-profile)# port-profile svm_vsd1_in
n1000v(config-port-profile)# vmware port-group
```

```
n1000v(config-port-profile)# switchport mode trunk
n1000v(config-port-profile)# switchport trunk allowed vlan 310-319
n1000v(config-port-profile)# virtual-service-domain vsd1
n1000v(config-port-profile)# service-port inside default-action drop
n1000v(config-port-profile)# no shutdown
  state enabled
n1000v(config-port-profile)# port-profile svm_vsd1_out
n1000v(config-port-profile)# vmware port-group
n1000v(config-port-profile)# switchport mode trunk
n1000v(config-port-profile)# switchport trunk allowed vlan 310-319
n1000v(config-port-profile)# virtual-service-domain vsd1
n1000v(config-port-profile)# service-port outside default-action drop
n1000v(config-port-profile)# no shutdown
```

Verifying the Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show virtual-service-domain name *vsd-name*	Displays a specific VSD configuration.
show virtual-service-domain brief	Displays a summary of all VSD configurations.
show virtual-service-domain interface	Displays the interface configuration for all VSDs.
module vem *module_number* **execute vemcmd show vsd**	Displays the VEM VSD configuration by sending the command to the VEM from the remote Cisco Nexus 1000V.
module vem *module_number* **execute vemcmd show vsd ports**	Displays the VEM VSD ports configuration by sending the command to the VEM from the remote Cisco Nexus 1000V.

Example: show virtual-service-domain name vsd_name

```
switch# show virtual-service-domain name vsd1
Default Action: drop

_____________________________
Interface        Type

_____________________________
Vethernet1       Member
Vethernet2       Member
Vethernet3       Member
Vethernet6       Member
Vethernet7       Inside
Vethernet8       Outside

switch#
```

Example: show virtual-service-domain brief

```
switch# show virtual-service-domain brief
Name   vsd-id    default action    in-ports   out-ports   mem-ports    Modules with
                                                                      VSD Enabled
zone    1         forward           1          1           2            4
switch#
```

Example: show virtual-service-domain interface

```
switch# show virtual-service-domain interface
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Name            Interface            Type      Status
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
vsd1            Vethernet1           Member    Active
vsd1            Vethernet2           Member    Active
vsd1            Vethernet3           Member    Active
vsd1            Vethernet6           Member    Active
vsd1            Vethernet7           Inside    Active
vsd1            Vethernet8           Outside   Active
vsd2            Vethernet9           Inside    Active
vsd2            Vethernet10          Outside   Active
switch#
```

Example: module module_number execute vemcmd show vsd

```
switch# module vem 4 execute vemcmd show vsd
ID Def_Act ILTL OLTL NMLTL State Member LTLs
1 FRWD 51 50 1 ENA 49
switch#
```

module module_number execute vemcmd show vsd ports

```
switch# module vem 4 execute vemcmd show vsd ports
LTL IfIndex VSD_ID VSD_PORT_TYPE
49 1c000010 1 REGULAR
50 1c000040 1 OUTSIDE
51 1c000030 1 INSIDE
switch#
```

Configuration Examples for VSD

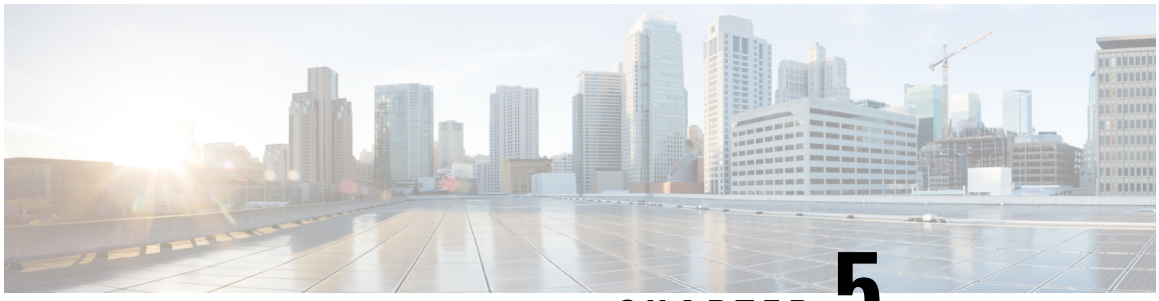
The following example shows how to configure VSD.

```
port-profile vsd1_member
  vmware port-group
  switchport access vlan 315
  virtual-service-domain vsd1
  no shutdown
  state enabled
port-profile svm_vsd1_in
  vmware port-group
  switchport mode trunk
  switchport trunk allowed vlan 310-319
  virtual-service-domain vsd1
  service-port inside default-action drop
  no shutdown
  state enabled
port-profile svm_vsd1_out
  vmware port-group
  switchport mode trunk
  switchport trunk allowed vlan 310-319
  virtual-service-domain vsd1
  service-port outside default-action drop
  no shutdown
```

Feature History for VSD

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
VSD	4.0(4)SV1(2)	This feature was introduced.



CHAPTER 5

Configuring AAA

This chapter contains the following sections:

Information About AAA

AAA Security Services

Based on a user ID and password combination, authentication, authorization, and accounting (AAA) is used to authenticate and authorize users. A key secures communication with AAA servers. AAA supports IPv4 and IPv6 addresses.

In many circumstances, AAA uses protocols such as RADIUS or TACACS+ 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.

Although AAA is the primary (and recommended) method for access control, additional features for simple access control are available outside the scope of AAA, such as local username authentication, line password authentication, and enable password authentication. However, these features do not provide the same degree of access control that is possible by using AAA.

Separate AAA configurations are made for the following services:

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

The following table provides the authentication commands:

AAA Service Configuration Option	Related Command
Telnet or SSH login	aaa authentication login default
Console login	aaa authentication login console

Authentication

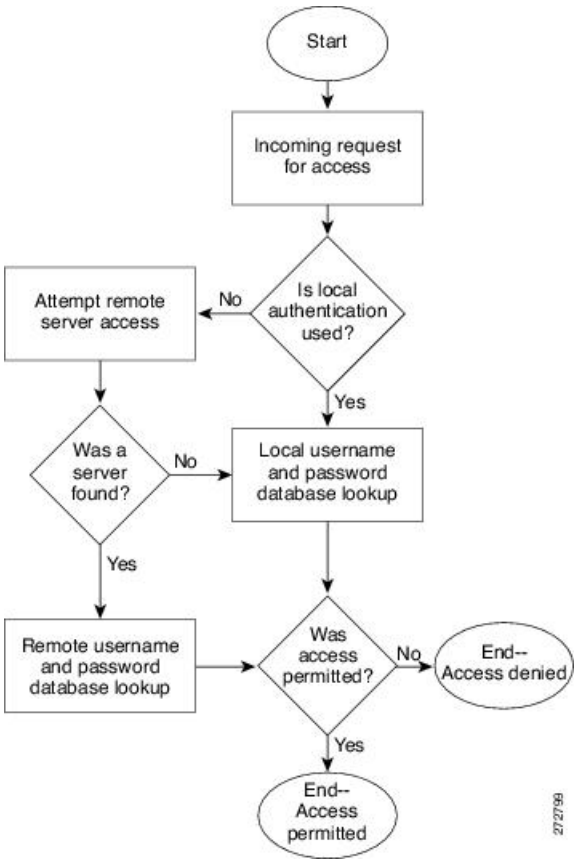
Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and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

Authentication is accomplished as follows:

Authentication Method	Description
Local database	Authenticates the following with a local lookup database of usernames or passwords: • Console login authentication • User login authentication • User management session accounting
Remote RADIUS or TACACS+ server	Authenticates the following with a local lookup database of usernames or passwords: • Console login authentication • User login authentication • User management session accounting
None	Authenticates the following with only a username: • Console login authentication • User login authentication • User management session accounting

The following figure shows a flowchart of the authentication process.

Figure 2: Authenticating User Login



Note This diagram is applicable only to username password SSH authentication. It does not apply to public key SSH authentication. All username password SSH authentication goes through AAA.

Authorization

Authorization restricts the actions that a user is allowed to perform. It 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

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.

Accounting tracks and maintains a log of every SVS management session. 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.

AAA Server Groups

Remote AAA server groups can provide failovers if one remote AAA server fails to respond, which means that if the first server in the group fails, the next server in the group is tried until a server responds. Multiple server groups can provide failovers for each other in this same way.

If all remote server groups fail, the local database is used for authentication.

Prerequisites for AAA

- At least one TACACS+ or RADIUS server is IP reachable
- The VSM is configured as an AAA server client.
- A shared secret key is configured on the VSM and the remote AAA server.

Guidelines and Limitations

The Cisco Nexus 1000V does not support usernames that have all numeric characters and does not create local usernames that have all numeric characters. If a username that has all numeric characters already exists on an AAA server and is entered during login, the Cisco Nexus 1000V does authenticate the user.

AAA Default Settings

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled

Configuring AAA

Configuring a Login Authentication Method

If authentication is to be done with TACACS+ server group(s), you must have already added the group(s).

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login** {**console** | **default**} {**group** *group-list* [**none**] | **local** | **none**}	Configures the console or default login authentication method. the keywords and arguments are as follows: • **group**—Specifies that authentication is done by server group(s). • *group-list*—List of server group names separated by spaces. • **none**— Specifies no authentication. • **local**—Specifies that the local database is used for authentication. **Note** Local is the default and is used when no methods are configured or when all the configured methods fail to respond. • **none**—Specifies that authentication is done by username.
Step 3	Required: switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configured login authentication method.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a login authentication method:

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

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

Enabling Login Authentication Failure Messages

You can enable the login authentication failure message to display if the remote AAA servers do not respond.

The following is the Login Authentication Failure message:

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

Example

This example shows how to enable login authentication failure messages:

```
switch# configure terminal
switch(config)# aaa authentication login error-enable
switch(config)# exit
switch# show aaa authentication login error-enable
enabled
```

Verifying the AAA Configuration

Use the following commands to verify the configuration:

Command	Purpose
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.

Example: show aaa authentication

```
switch# show aaa authentication login error-enable
disabled
switch#
```

Example: show running config aaa

```
switch# show running-config aaa all
version 4.0(1)
aaa authentication login default local
aaa accounting default local
no aaa authentication login error-enable
no aaa authentication login mschap enable
no radius-server directed-request
no snmp-server enable traps aaa server-state-change
no tacacs-server directed-request
switch#
```

Example: show startup-config aaa

```
switch# show startup-config aaa
version 4.0(1)
```

Configuration Examples for AAA

The following is an AAA configuration example:

```
aaa authentication login default group tacacs
aaa authentication login console group tacacs
```

Feature History for AAA

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
AAA	4.0(4)SV1(1)	This feature was introduced.

Secure Login Enhancements

Starting with Cisco Nexus 1000V for VMware vSphere Release 5.2(1)SV3(4.1a), you can configure login parameters to enhance secure login to Cisco Nexus 1000V switches.

Configuring Login Parameters

Use this task to configure your Cisco Nexus 1000V device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

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

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

Procedure

Step 1 **configure terminal**

Example:

```
Switch# configure terminal
```

Enters global configuration mode.

Step 2 **[no] login block-for** *seconds* **attempts** *tries* **within** *seconds*

Example:

```
Switch(config)# login block-for 100 attempts 2 within 100
```

Configures your Cisco NX-OS device for login parameters that help you detect DoS attack.

Note This command must be issued before any other login command can be used.

Step 3 **[no] login quiet-mode access-class** {*acl-name* | *acl-number*}

Example:

```
Switch(config)# login quiet-mode access-class myacl
```

(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.

Step 4 **exit**

Example:

```
Switch(config)# exit
```

Exits to privileged EXEC mode.

Step 5 **show login failures**

Example:

```
Switch# show login failure
```

Displays login parameters.

- **failures** - Displays information related to failed login attempts.

Configuration Examples for Login Parameters

Setting Login Parameters Example

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

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

Showing Login Parameters Example

The following sample output from the **show login** command verifies that secure login parameters have been specified:

```
Switch# show login

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

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

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

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

```
Switch# show login failures

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

```
admin                                    pts/0   ws.cisco.com   login
        Wed Jun 10 04:56:19 2015
------------------------------------------------------------------------------
```

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

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

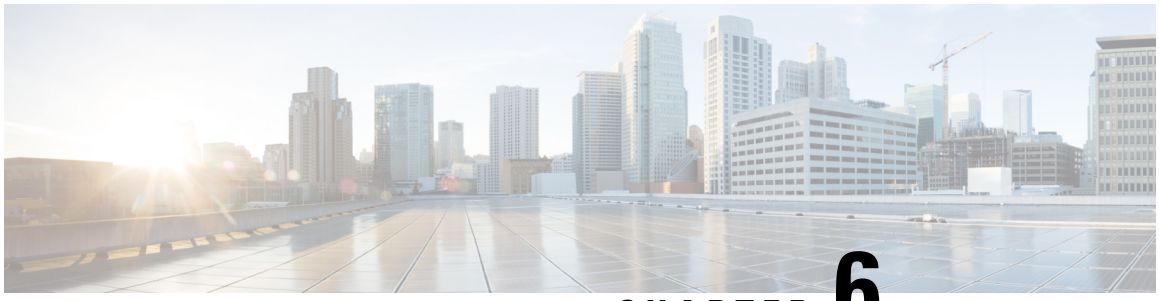
The following example shows how to clear the failed login attempts using the clear command:

```
Switch# clear login failures

This command is provided to clear statistics about failure details

Usage:

Nexus 1000v# sh login failures

Information about last 20 login failure's with the device.
-------------------------------------------------------------------------------------------------------
Username                                 Line    SourceIPAddr    Appname          TimeStamp
-------------------------------------------------------------------------------------------------------
admin                                    ssh     10.78.184.85    login            Mon Feb 18
 07:38:16 2019
admin                                    ssh     10.78.184.85    login            Mon Feb 18
 07:38:18 2019
-------------------------------------------------------------------------------------------------------
Nexus 1000v#

Nexus 1000v# clear login failures
Nexus 1000v#
Nexus 1000v# sh login failures
```

Guidelines and Limitations

Follow these usage guidelines and limitations while configuring Secure Login Enhancements:

- When the Quite mode is activated and login access is blocked for SSH and Telnet with ACLs, existing login sessions are also stopped. This behavior is consistent with the regular ACL behavior as applied to any interface handling traffic.
- Ensure that ACLs have last entries as “permit ip any any” in order to allow any other permitted protocol traffic to pass through the management interface, other than those handled by ACL entries. Default policy otherwise is to deny such additional IP traffic.
- PNSC access to VSM could get blocked due to ACL. To avoid this issue, configure secure login on VSM such that https access between VSM and PNSC is possible bidirectionally. Corresponding port to be opened for this purpose is 443.
- Secure login feature does not work together with ACLs directly configured with management interface (mgmt0) for VSM. Both are mutually exclusive configurations.



CHAPTER 6

Configuring RADIUS

This chapter contains 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.

Note RADIUS supports IPv4 and IPv6 addresses.

RADIUS Network Environments

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

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

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and

end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.

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

RADIUS Operation

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

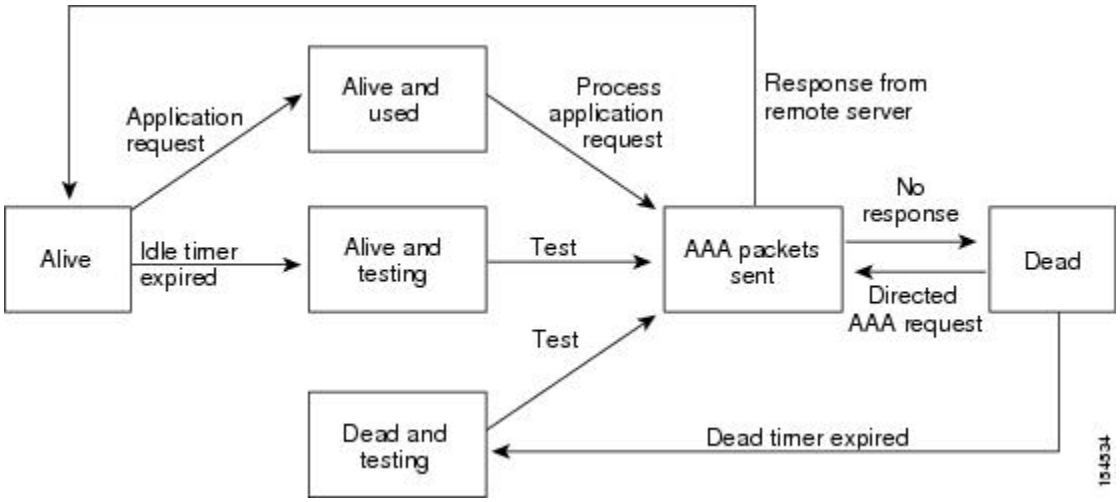
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. Unresponsive RADIUS servers are marked as dead and are not sent AAA requests. Dead RADIUS servers are periodically monitored and returned 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 an error message is displayed indicating that a failure is taking place.

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.

Figure 3: Radius Server States



Vendor-Specific Attributes

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

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

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

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

The following VSA protocol options are supported:

- 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 following attributes are supported:

- 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, the value field would be “network-operator”. This attribute, which the RADIUS server sends in the VSA portion of the

Access-Accept frames, can be used only with the shell protocol value. The following examples show the roles attribute as supported by Cisco Access Control System (ACS):

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

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

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

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

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

If you are using Cisco ACS and intend to use the same ACS group for both Cisco Nexus 1000V and Cisco UCS authentication, use the following roles attribute:

```
cisco-av-pair*shell:roles="network-admin admin"
```

Note When you specify a VSA as shell:roles*"network-operator " or "shell:roles*\"network-operator \"", 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).

Prerequisites for RADIUS

- You already know the RADIUS server IP addresses or hostnames.
- You already know the key(s) used to secure RADIUS communication in your network.
- The device is already configured as a RADIUS client of the AAA servers.

Guidelines and Limitations

You can configure a maximum of 64 RADIUS servers.

Default Settings

Table 4: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1

Parameters	Default
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

Configuring RADIUS Server Hosts

You can configure the IP address (IPv4 or IPv6) or the hostname for each RADIUS server to be used for authentication. You should know the following information:

- You can configure up to 64 RADIUS servers.
- All RADIUS server hosts are automatically added to the default RADIUS server group.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*}	Defines the IP address or hostname for the RADIUS server, or the RADIUS server Domain Name Server (DNS) name. host-name—The *host-name* argument is alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a RADIUS server host using IPv4 address:

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

Configuring the Global RADIUS Key

You can configure the key that is used by all RADIUS servers to authenticate with the Cisco Nexus 1000V.

Before you begin

- Log in to the CLI in EXEC mode.
- You must know the global key that is used for RADIUS server authentication.

Procedure

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

Example

This example shows how to configure a global RADIUS key:

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

Configuring a RADIUS Accounting Server

You can configure a server to perform accounting functions.

By default, RADIUS servers are used for both accounting and authentication.

Before you begin

- Logged in to the CLI in EXEC mode.
- Know the destination UDP port number for RADIUS accounting messages.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*	Associates a specific host with the UDP port that receives RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**	Designates the specific RADIUS host as an accounting server. The default is both accounting and authentication.
Step 4	switch(config)# **exit**	Returns you to EXEC mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a RADIUS accounting server using IPv4 address:

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

Configuring RADIUS Server Groups

You can configure a RADIUS server group whose member servers share authentication functions.

The servers in the group are tried in the same order in which you configure them

Before you begin

- Log in to the CLI in EXEC mode.
- Know that all servers in a RADIUS server group must belong to the RADIUS protocol.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration mode for that group. The group-name argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.
Step 3	switch(config-radius)# **server** {*ipv4-address* | *ipv6-address* | *server-name*}	Configures the RADIUS server as a member of the RADIUS server group. **Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) switch(config-radius)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-radius)# **use-vrf** *vrf-name*	Specifies the VRF to use to contact the servers in the server group
Step 6	(Optional) switch(config-radius)# **source-interface** {*interface-type*} {*interface-number*}	Specifies a source interface to be used to reach the RADIUS server. The interface types and interface numbers are defines as follows: • loopback—Virtual interface number from 0 to 1023 • mgmt—Management interface 0 • null—Null interface 0 • port-channel—Port channel number from 1 to 4096

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

Example

This example shows how to configure a RADIUS server group using IPv4 address:

```
switch# configure terminal
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# deadtime 30
switch(config-radius)# use-vrf vrf1
switch(config-radius)# source-interface mgmt0
switch(config-radius)# show radius-server group
total number of groups:2

following RADIUS server groups are configured:
        group Radserver:
                server: 10.10.1.1
                deadtime is 30
        group test:
                deadtime is 30
switch(config-radius)# copy running-config startup-config
```

Enabling RADIUS Server-Directed Requests

You can allow users to designate the RADIUS server to send their authentication request to. This process is called a directed request.

If you enable this option, a user can log in as username@vrfname:hostname, where *vrfname* is the virtual routing and forwarding (VRF) to use and *hostname* is the name of a configured RADIUS server.

Directed requests are disabled by default.

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **radius-server directed-request**	Enables directed requests. The default is disabled.
Step 3	switch(config)# **exit**	Returns to EXEC mode.
Step 4	(Optional) switch(config)# **show radius-server directed-request**	Displays the directed request configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable a RADIUS server-directed request:

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

Setting a Global Timeout for All RADIUS Servers

You can configure the global timeout interval that specifies how long to wait for a response from a RADIUS server before declaring a timeout failure.

The timeout specified in the “Setting the Timeout Interval for a Single RADIUS Server” section overrides the global RADIUS timeout.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

Example

This example shows how to set a global timeout for all RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server timeout 101
switch(config-radius)# exit
switch(config-radius)# show radius-server
switch(config-radius)# copy running-config startup-config
```

Configuring a Global Retry Count for All RADIUS Servers

You can configure the maximum number of times to retry transmitting to a RADIUS server before reverting to local authentication.This setting is applied to all RADIUS servers.

By default, retransmission to a RADIUS server is only tried once before reverting to local authentication.

You can increase the number of retries up to a maximum of five.

The retry count specified for a single RADIUS server in the “Configuring Retries for a Single RADIUS Server” section, overrides this global setting.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server retransmit***count*	Defines the number of retransmits allowed before reverting to local authentication. This global setting applies to all RADIUS servers. The default number of retransmits is 1 and the range is from 0 to 5.
Step 3	switch(config)# **exit**	Returns you to EXEC mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a global retry count for all RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server retransmit 31
switch(config)# exit
```

```
switch# show radius-server
switch# copy running-config startup-config
```

Setting a Timeout Interval for a Single RADIUS Server

You can configure how long to wait for a response from a RADIUS server before declaring a timeout failure.

The timeout specified for a single RADIUS server overrides the timeout defined in the "Setting the Global Timeout for All RADIUS Servers" section.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*	Specifies the timeout interval for the specified server. The default timeout interval is 5 seconds and the allowable range is from 1 to 60 seconds. **Note** The timeout specified for a single RADIUS server overrides the global RADIUS timeout.
Step 3	switch(config)# **exit**	Returns you to EXEC mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set a timeout interval for a single RADIUS server:

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

Configuring Retries for a Single RADIUS Server

You can configure the maximum number of times to retry transmitting to a RADIUS server before reverting to local authentication. This setting applies to a single RADIUS server and takes precedence over the global retry count.

Before you begin

Log in to the CLI in EXEC mode.

Know the following:

- By default, retransmission to a RADIUS server is only tried once before reverting to local authentication.
- You can increase the number of retries up to a maximum of five.
- The retry count specified for a single RADIUS server overrides the global setting made for all RADIUS servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. *ipv4/ipv6-address*—The IP address for the RADIUS server. *hostname*—The hostname for the RADIUS server. The hostname is alphanumeric, case sensitive, and has a maximum of 256 characters. *count*—The retransmission count. The default value is the global value. **Note** This retransmit count for a single RADIUS server overrides the global setting for all RADIUS servers.
Step 3	switch(config)# **exit**	Returns you to EXEC mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example configures retries for a single RADIUS server:

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

Configuring a RADIUS Accounting Server

You can configure a server to perform accounting functions.

By default, RADIUS servers are used for both accounting and authentication.

Before you begin

- Logged in to the CLI in EXEC mode.
- Know the destination UDP port number for RADIUS accounting messages.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*	Associates a specific host with the UDP port that receives RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**	Designates the specific RADIUS host as an accounting server. The default is both accounting and authentication.
Step 4	switch(config)# **exit**	Returns you to EXEC mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a RADIUS accounting server using IPv4 address:

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

Configuring a RADIUS Authentication Server

You can configure a server to perform authentication functions.

By default, RADIUS servers are used for both accounting and authentication.

Before you begin

- Log in to the CLI in EXEC mode.
- Know the destination UDP port number for RADIUS authentication messages.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port*	Associates a specific host with the UDP port that receives RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication**	Designates the specific RADIUS host as an authentication server. The default is both accounting and authentication.
Step 4	switch(config)# **exit**	Returns you to EXEC mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a RADIUS authentication server using IPv4 address:

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

Configuring Periodic RADIUS Server Monitoring

You can configure the monitoring of RADIUS servers.

The test idle timer specifies the interval of time that elapses before a test packet is sent to a non-responsive RADIUS server.

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

Note For security reasons, do not configure a username that is in the RADIUS database as a test username.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time***minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	switch(config)# **radius-server dead-time** *minutes*	Specifies the number of minutes to wait before sending a test packet to a RADIUS server that was declared dead. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	switch(config)# **exit**	Returns you to EXEC mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure periodic RADIUS server monitoring using IPv4 address:

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

Configuring the Global Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time to wait after declaring a RADIUS server 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.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

Example

This example shows how to configure the global dead-time interval:

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

Manually Monitoring RADIUS Servers or Groups

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

	Command or Action	Purpose
Step 2	switch# **test aaa server radius** {*ipv4-address* | *ipv6-address* | *server-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a RADIUS server to confirm availability.
Step 3	switch(config)# **test aaa group** *group-name username password*	Sends a test message to a RADIUS server group to confirm availability.

Example

This example shows how to manually monitor a RADIUS server or group using IPv4 address:

```
switch# configure terminal
switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Verifying the RADIUS Configuration

Use the following commands to verify the configuration.

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

Displaying RADIUS Server Statistics

Use the following command to display statistics for RADIUS server activity:

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

Configuration Example for RADIUS

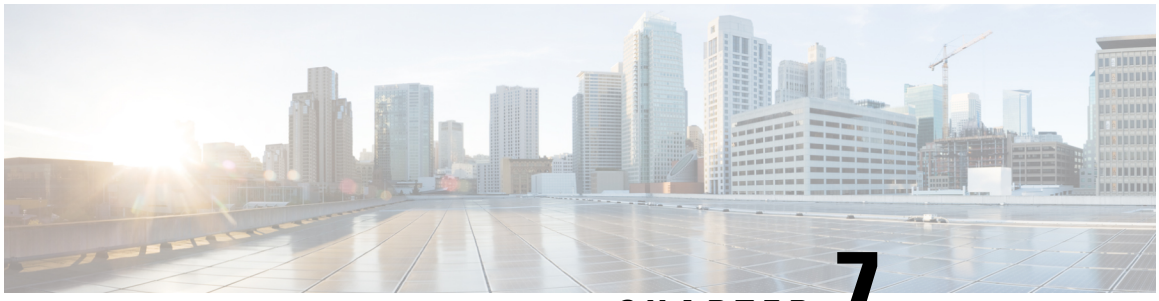
This example shows how to configure a global RADIUS key and a RADIUS server host key using IPv4 address:

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

Feature History for RADIUS

This table only includes updates for those release that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
RADIUS	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 7

Configuring TACACS+

This chapter contains the following sections:

Information About TACACS+

The TACACS+ security protocol provides centralized validation of users who are attempting to gain access to a device. TACACS+ services are maintained in a database on a TACACS+ daemon that is 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 device are available.

TACACS+ provides for separate authentication, authorization, and accounting services. The TACACS+ daemon provides each service 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. Centralized authentication is provided using the TACACS+ protocol.

Note TACACS+ security protocol supports IPv4 and IPv6 addresses.

TACACS+ Operation for User Login

The following sequence of events take place when you attempt to log in to a TACACS+ server using the Password Authentication Protocol (PAP):

1. When a connection is established, the TACACS+ daemon is contacted 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 might include prompts for additional information, such as your mother's maiden name.

2. The TACACS+ daemon provides one of the following responses:
   1. ACCEPT—User authentication succeeds and service begins. If user authorization is needed, authorization begins.
   2. REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   3. ERROR—An error occurred at some time during authentication either at the daemon or in the network connection. If an ERROR response is received, the device tries to use an alternative method for authenticating the user.

   If further authorization is required after authentication, the user also undergoes an additional authorization phase. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the TACACS+ daemon is contacted 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, access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

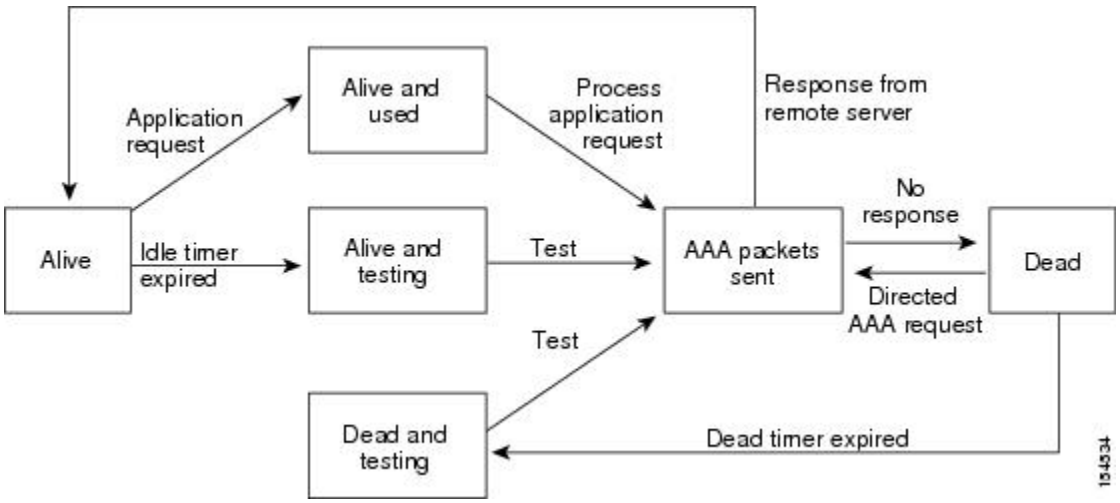
You must configure the TACACS+ preshared key to authenticate to the TACACS+ server. A preshared key is a secret text string shared between the device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not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

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

TACACS+ Server Monitoring

Unresponsive TACACS+ servers are marked as dead and are not sent AAA requests. Dead TACACS+ servers are periodically monitored and brought back alive once they respond. This process confirms that a TACACS+ server is in a working state before real AAA requests are sent its way. The following figure shows how a TACACS+ server state change generates a Simple Network Management Protocol (SNMP) trap and an error message showing the failure before it impacts performance.

Figure 4: 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.

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. The separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use TACACS+ servers for authentication, the TACACS+ protocol directs the TACACS+ server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported:

- 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 following attributes are other supported:

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

Prerequisites for TACACS+

- Obtain the IP addresses or hostnames for the TACACS+ servers.
- Obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus 1000V is configured as a TACACS+ client of the AAA servers.
- You have already configured AAA, including remote TACACS+ authentication.

Guidelines and Limitations for TACACS+

- You can configure a maximum of 64 TACACS+ servers.
- The logging level for TACACS + must be set to 5.
- We recommend that you configure the dead-time interval if more than six servers are configured in a group. If you must configure more than six servers, make sure to set the dead-time interval to a value greater than 0 and enable dead server monitoring by configuring the test username and test password.
- TACACS+ type-6 password encryption is applicable in the following order:
  1. feature configuration
  2. creating the master key
  3. configuration of the TACAC+ server key

Note You will not be able to achieve a proper TACAC+ type-6 password encryption if the above order is not followed.

Default Settings for TACACS+

Parameters	Default
TACACS+	Disabled

Parameters	Default
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring TACACS+

The following flowchart guides you through the TACACS+ configuration process.

Note Be aware that the Cisco Nexus 1000V commands might differ from the Cisco IOS commands.

Figure 5: Configuring TACACS+ Flowchart

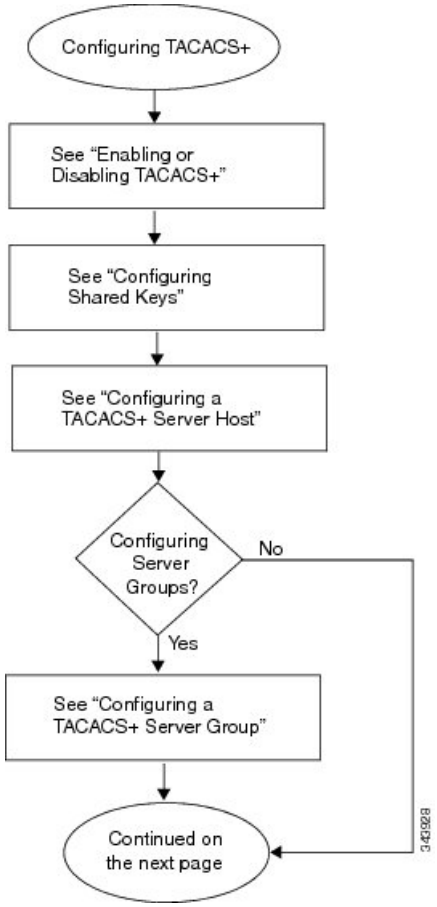


Figure 6: Configuring TACACS+ Flowchart (continued)

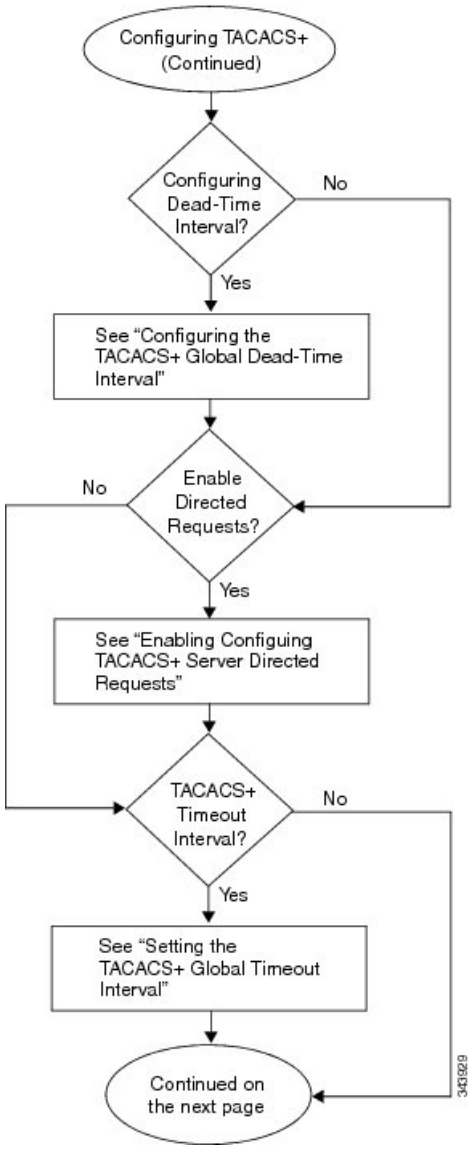
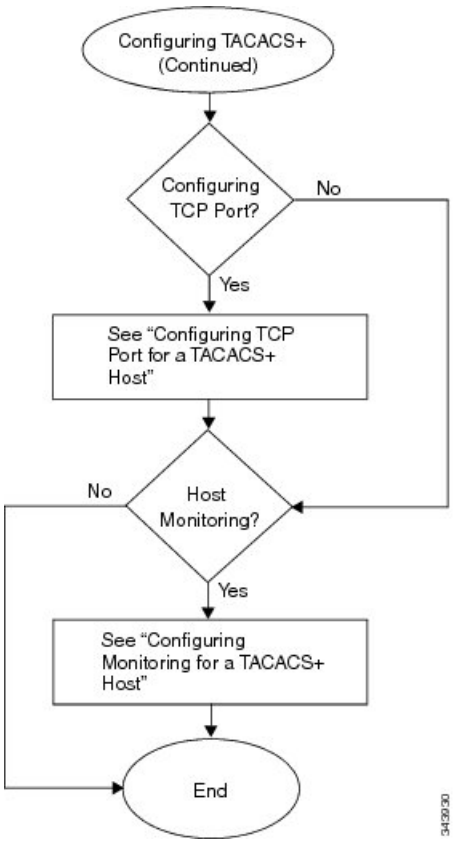


Figure 7: Configuring TACACS+ Flowchart (continued)



Enabling or Disabling TACACS+

By default, TACACS+ is disabled. You must explicitly enable the TACACS+ feature to access the configuration and verification commands that support TACACS+ authentication.

⚠

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **tacacs+ enable**	Enables or disables TACACS+.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.

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

Example

This example shows how to enable TACACS+:

```
switch# configure terminal
switch(config)# tacacs+ enable
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Shared Keys

By default, no global key is configured.

You can configure the following:

- The global key or a secret text string that is shared between the Cisco Nexus 1000V and all TACACS+ server hosts
- The key or secret text string that is shared between the Cisco Nexus 1000V and a single TACACS+ server host

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Know the key for the TACACS+ server host(s).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode. Do one of the following: • To configure a global key for all TACACS+ server hosts, continue to the next step. • To configure a key for a single TACACS+ server host, go to Step 3.

	Command or Action	Purpose
Step 2	switch(config)# **tacacs-server key** [**0** | **7**] *global_key*	Designates the global key shared between the Cisco Nexus 1000V and the TACACS+ server hosts. • **0**—Specifies a clear text string (key) to follow. This is the default. • **7**—Specifies an encrypted string (key) to follow. • *global_key*— String of up to 63 characters. By default, no global key is configured. Go to Step 4.
Step 3	switch(config)# **tacacs-server host** { *ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *shared_key*	Designates the key shared between the Cisco Nexus 1000V and this specific TACACS+ server host. **0**—Specifies a clear text string (key) to follow. This is the default. **7**—Specifies an encrypted string (key) to follow. *global key*—String of up to 63 characters. This shared key is used instead of the global shared key.
Step 4	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 5	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration. **Note** The global shared key is saved in encrypted form in the running configuration. To display the key, use the **show running-config** command.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a shared key:

```
switch# configure terminal
switch(config)# tacacs-server key 0 QsEFtkI#
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
```

```
timeout value:5
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
switch# copy running-config startup-config
```

Configuring a TACACS+ Server Host

All TACACS+ server hosts are added to the default TACACS+ server group.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the shared key.
- Know the IP addresses or the hostnames for the remote TACACS+ server hosts.

Procedure

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

Example

This example shows how to configure a TACACS+ server host using IPv4 address:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2
switch(config)# exit
switch# show tacacs-server
timeout value:5
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
```

```
                available on port:49
switch# copy running-config startup-config
```

Configuring a TACACS+ Server Group

You can configure a TACACS+ server group whose member servers share authentication functions.

After you configure the TACACS+ server group, the server members are tried in the same order in which you configured them.

A TACACS+ server group can provide a failover if one server fails to respond. If the first server in the group fails, the next server in the group is tried until a server responds. Multiple server groups can provide failovers for each other in this same way.

Before you begin

- Log in to the CLI in EXEC mode.
- Know that all servers added to a TACACS+ server group use the TACACS+ protocol.
- Configure the preshared keys.
- Enable TACACS+ for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa group server tacacs+** *group-name*	Creates a TACACS+ server group with the specified name and places you into the TACACS+ configuration mode for that group.
Step 3	switch(config-tacacs+)# **server** {*ipv4-address* | *ipv6-address* | *host-name*}	Configures the TACACS+ server hostname or IP address as a member of the TACACS+ server group. If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.
Step 4	(Optional) switch(config-tacacs+)# **deadtime** *minutes*	Configures the monitoring dead time for this TACACS+ group. 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.

	Command or Action	Purpose
Step 5	(Optional) switch(config-tacacs+)# **use-vrf** *vrf-name*	Specifies the virtual routing and forwarding instance (VRF) to use to contact this server group
Step 6	(Optional) switch(config-tacacs+)# **source-interface** {*interface-type*} {*interface-number*}	Specifies a source interface to be used to reach the TACACS+ server. • **loopback**—Virtual interface number from 0 to 1023 • **mgmt**—Management interface 0 • **null**—Null interface 0 • **port-channel**—Port channel number from 1 to 4096
Step 7	(Optional) switch(config-tacacs+)# **show tacacs-server groups**	Displays the TACACS+ server group configuration
Step 8	(Optional) switch(config-tacacs+)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a TACACS+ server group using IPv4 address:

```
switch# config terminal
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# deadtime 30
switch(config-tacacs+)# use-vrf management
switch(config-tacacs+)# source-interface mgmt0
switch(config-tacacs+)# show tacacs-server groups
total number of groups:1

following TACACS+ server groups are configured:
        group TacServer:
                server 10.10.2.2 on port 49
                deadtime is 30
                vrf is management
switch# copy running-config startup-config
```

Enabling TACACS+ Server-Directed Requests

You can designate which TACACS+ server to send an authentication request to. This process is called a directed request.

When directed requests are enabled, you 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 User-specified logins are supported only for Telnet sessions.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Enables use of directed requests for specifying the TACACS+ server to send an authentication request to when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 4	(Optional) switch(config)# **show tacacs-server directed-request**	Displays the TACACS+ directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to enable a TACACS+ server-directed request:

```
switch# config terminal
switch(config)# tacacs-server directed-request
switch(config)# exit
switch# show tacacs-server directed-request
enabled
switch# copy running-config startup-config
```

Setting the TACACS+ Global Timeout Interval

You can set the interval in seconds that the Cisco Nexus 1000V waits for a response from any TACACS+ server before declaring a timeout.

The timeout specified for an individual TACACS+ server overrides the global timeout interval. To set the timeout for an individual server.

Before you begin

- Log in to the CLI in EXEC mode.

- Enable TACACS+ for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server timeout** *seconds*	Specifies the interval in seconds that the Cisco Nexus 1000V waits for a response from a server. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds.
Step 3	switch(confi)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 4	(Optional) switch(config)# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch(confi)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to set a TACACS+ timeout interval:

```
switch# configure terminal
switch(config)# tacacs-server timeout 10
switch(config)# exit

switch# n1000v# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
switch# copy running-config startup-config
```

Setting a Timeout Interval for an Individual TACACS+ Host

You can set the interval in seconds that the Cisco Nexus 1000V waits for a response from a specific TACACS+ server before declaring a timeout. This setting is configured per TACACS+ host.

The timeout setting for an individual TACACS+ server overrides the global timeout interval.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*	Specifies the timeout interval for a specific server. The default is the global timeout interval.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 4	(Optional) switch(config)# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to set a timeout interval for an individual TACACS+ host using IPv4 address:

```
switch# config terminal
switch(config)#  tacacs-server host 10.10.2.2 timeout 10
switch(config)# exit
switch# n1000v# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:49
                timeout:10
switch# copy running-config startup-config
```

Configuring the TCP Port for a TACACS+ Host

You can configure a TCP port other than port 49 (the default for TACACS+ requests).

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port*	Specifies the TCP port to use. The port range is from 1 to 65535. The default is 49.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 4	(Optional) switch(config)# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to configure the TCP port for a TACACS+ host using IPv4 address:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2 port 2
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:0
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Configuring Monitoring for a TACACS+ Host

You should know the following information:

- The idle timer specifies how long a TACACS+ server should remain idle (receiving no requests) before sending it a test packet.
- The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not done.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name* } **test** {**idle-time***minutes* |**password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time***minutes*]] }	Configures server monitoring. The keywords and arguments are as follows: • **username**—Specifies that the default is test. **Note** To protect network security, we recommend that you assign a username that is not already in the TACACS+ database. • **password**—Specifies that the default is test. • **idle-time**—The default is 0 minutes. 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	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the duration of time in minutes before checking 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	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 5	(Optional) switch(config)# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to configure monitoring for a TACACS+ host using IPv4 address:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2 test username pvk2 password a3z9yjqz7 idle-time
 3
switch(config)# tacacs-server dead-time 5
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:5
```

```
total number of servers:1

following TACACS+ servers are configured:
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Configuring the TACACS+ Global Dead-Time Interval

You can configure the interval to wait before sending a test packet to a previously unresponsive server.

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 time per group.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable TACACS+ for authentication.
- Configure the TACACS+ server.

Procedure

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

Example

This example shows how to configure the TACACS+ global dead-time interval:

```
switch# configure terminal
switch(config)#  tacacs-server deadtime 5
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:5
total number of servers:1

following TACACS+ servers are configured:
```

```
        10.10.2.2:
                available on port:2
                timeout:10
switch# copy running-config startup-config
```

Displaying Statistics for a TACACS+ Host

Use the following command to display statistics for a TACACS+ host.

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

Configuration Example for TACACS+

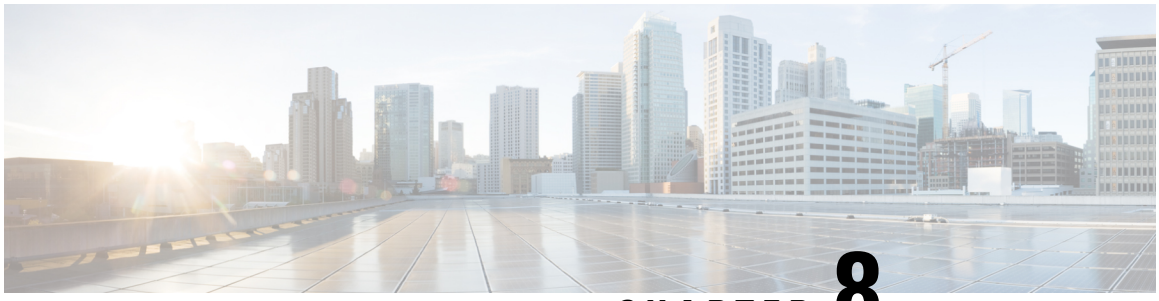
This example shows a TACACS+ configuration:

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

Feature History for TACACS+

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
TACACS+	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 8

Configuring SSH

This chapter contains the following sections:

Information About SSH

SSH Server

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

TACACS+ user authentication and locally stored usernames and passwords are supported for SSH.

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 secure, encrypted connection to any device that runs the SSH server. This connection provides an encrypted outbound connection. With authentication and encryption, the SSH client produces secure communication over an insecure network.

The SSH client works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communication. 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 correct version before enabling the SSH service. 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, an RSA key that uses 1024 bits is generated.

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.

Prerequisites for SSH

- Configure IP on a Layer 3 interface, out-of-band on the mgmt 0 interface or inband on an Ethernet interface. SSH supports both IPv4 and IPv6 addresses.
- Before enabling the SSH server, obtain the SSH key.

Guidelines and Limitations for SSH

- Only SSH version 2 (SSHv2) is supported.
- SSH is enabled by default.
- Cisco NX-OS commands might differ from the Cisco IOS commands.

Default Settings

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

Configuring SSH

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements.

The default SSH server key is an RSA key that is generated using 1024 bits.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables SSH.
Step 3	switch(config)# **ssh key** {**dsa**[**force**] | **rsa** [*bits*[**force**]]}	Generates the SSH server key The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 4	switch(config)# **feature ssh**	Enables SSH.
Step 5	(Optional) switch# **show ssh key**	Displays the SSH server keys.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to generate SSH server keys:

```
switch# configure terminal
switch(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
switch(config)# ssh key dsa force
generating dsa key(1024 bits).....
.
generated dsa key
n1000v(config)# feature ssh
n1000v(config)# show ssh key
*************************************
rsa Keys generated:Sun Jul 27 15:18:46 2008

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXK
fVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6Uwa
```

```
GID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+f
FzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4
GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==

bitcount:2048
fingerprint:
fd:ca:48:73:b9:ee:e7:86:9e:1e:40:46:f1:50:1d:44
************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
************************************
```

Configuring a User Account with a Public Key

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

- OpenSSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Configuring an OpenSSH Key

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

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

- OpenSSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Before you begin

- Log in to the CLI in EXEC mode
- Generate an SSH public key in OpenSSH format
- Have an existing user account

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in OpenSSH format with an exiting user account. To create a user account use the **username** *name* **password** *pwd* command.
Step 3	switch(config)# **exit**	Exits global configuration mode and returns to EXEC mode.
Step 4	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure an openSSH key:

```
switch# configure terminal
switch(config)# username user1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyK
cb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5aw
fVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8
QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuD
YSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby
/ZWx/clTYiXQR1X1VfhQ==
switch(config)# exit
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdH
Ha/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6
/n3FVroyRwupMki6mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EI
r/0XIP1mqTsrqTsmjZ2vLk+fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m
9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1Vf
hQ==
switch# copy running-config startup-config
```

Configuring IETF or PEM Keys

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

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

- OpenSSH format
- IETF SECSH format

- Public Key Certificate in PEM format

Before you begin

- Log in to the CLI in EXEC mode
- Generate an SSH public key in one of the following formats:
  - IETF SECSH format
  - Public Key Certificate in PEM format

Procedure

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

Example

This example shows how to configure an SSH public key in an IETF SECSH format:

```
switch# copy tftp://10.78.1.10/secsh_file.pub bootflash:secsh_file.pub vrf management
Trying to connect to tftp server......
Connection to server Established.
|
TFTP get operation was successful
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:user2
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/
ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6
mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+
fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJN
```

```
U1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==
switch# copy running-config startup-config
```

Starting SSH Sessions

You can start SSH sessions using IP to connect to remote devices.

Before you begin

- Log in to the CLI in EXEC mode.
- Obtain the hostname and, if needed, the username, for the remote device.
- Enable the SSH server on the remote device

Procedure

	Command or Action	Purpose
Step 1	switch# **ssh** [**root@**] {*ip address* | *hostname* } [**vrf** *vrf-name*] or switch# **ssh6** [**root@**] {*ip address* | *hostname* } [**vrf** *vrf-name*]	Creates an SSH IPv4 or IPv6 session to a remote device using IP. The default virtual routing and forwarding (VRF) instance is the default VRF.

Example

This example shows how to start an SSH session:

```
switch# ssh root@172.28.30.77
root@172.28.30.77's password:
Last login: Sat Jul 26 11:07:23 2008 from 171.70.209.64
```

Clearing SSH Hosts

You can clear from your account the list of trusted SSH servers that were added when you downloaded a file from a server using SCP or SFTP, or when you started an SSH session to a remote host.

Procedure

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

Disabling the SSH Server

You can disable the SSH server to prevent SSH access to the switch. By default, the SSH server is enabled.

If you disable SSH, you must first generate an SSH server key before you can enable it again.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

Example

This example shows how to disable the SSH server:

```
switch# configure terminal
switch(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
switch(config)#  show ssh server
ssh is not enabled
switch(config)# copy running-config startup-config
```

Deleting SSH Server Keys

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

If you disable SSH, you must first generate an SSH server key before you can enable it again.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

Example

This example shows how to delete an SSH server key:

```
switch# configure terminal
switch(config)# no feature ssh
switch(config)# no ssh key rsa
switch(config)# show ssh key
**************************************
rsa Keys generated:Sun Jul 27 15:18:46 2008

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXK
fVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mWoM6Uwa
GID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+f
FzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4
GVc6sMJNU1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==

bitcount:2048
fingerprint:
fd:ca:48:73:b9:ee:e7:86:9e:1e:40:46:f1:50:1d:44
**************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
**************************************
mcs-srvr43(config)# no ssh key rsa
mcs-srvr43(config)# show ssh key
**************************************
could not retrieve rsa key information
**************************************
dsa Keys generated:Sun Jul 27 15:20:12 2008

ssh-dss AAAAB3NzaC1kc3MAAACBALpdxLjXNS/jcCNY+F1QZV9HegxBEb0DMUmq9bSq2N+KAcvHllEh
GnaiHhqarOlcEKqhLbIbuqtKTCvfa+Y1hBIAhWVjg1UR3/M22jqxnfhnxL5YRc1Q7fcesFax0myayAIU
nXrkO5iwv9XHTu+EInRc4kJ0XrG9SxtLmDe/fi2ZAAAAFQDbRabAjZa6GfDpwjXw5smRhrElJwAAAIEA
r50yi3hHawNnb5qgYLXhN+KA8XJF753eCWHtMw7NR8fz6fjQ1R2J97UjjGuQ8DvwpGeNQ5S+AuIo0rGq
svdg7TTecBcbgBOnR7Fs2+W5HiSVEGbvj1xaeK8fkNE6kaJumBB343b8Rgj0G97MP/os1GfkEqmX9glB
0IOM2mgHHyoAAACAfRir27hHy+fw8CxPlsK0R6cFhxYyd/qYYogXFKYIOPxpLoYrjqODeOFThU7TJuBz
aS97eXiruzbffHwzUGfXgmQT5o9IMZRTClWPA/5Ju4O9YABYHccUghf0W+QtgGOT6FOSvBh8uOV0kcHC
GMJAP8omphauZJlc+wgFxhnkyh4=

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
**************************************
mcs-srvr43(config)# no ssh key dsa
mcs-srvr43(config)# show ssh key
**************************************
could not retrieve rsa key information
**************************************
could not retrieve dsa key information
**************************************
```

```
no ssh keys present. you will have to generate them
************************************
```

Clearing SSH Sessions

You can clear SSH sessions from the device.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user SSH session.
Step 3	(Optional) switch# **show users**	Displays user session information.

Example

This example shows how to clear an SSH session:

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/0        Jul 28 09:49 00:02       28556 (10.21.148.122)
admin    pts/1        Jul 28 09:46   .         28437 (::ffff:10.21.148.122)*
switch# clear line 0
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/1        Jul 28 09:46   .         28437 (::ffff:10.21.148.122)*
mcs-srvr43(config)#
```

Verifying the SSH Configuration

Use the following commands to verify the configuration.

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.

Configuration Example for SSH

This example shows how to configure SSH with an OpenSSH key:

1. Disable the SSH server.

```
switch# configure terminal
switch(config)# no feature ssh
```

2. Generate an SSH server key.

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

3. Enable the SSH server.

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

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

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=
```

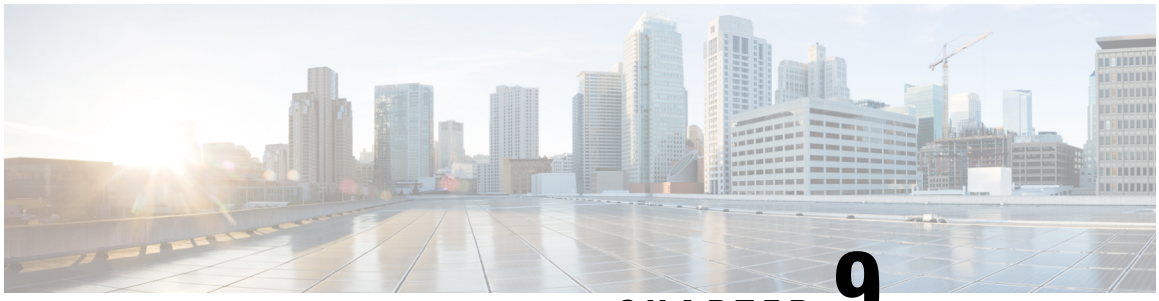
6. Save the configuration.

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

Feature History for SSH

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
SSH	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 9

Configuring Telnet

This chapter contains the following sections:

Information About the Telnet Server

The Telnet protocol enables you to set up 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 pass the keystrokes from one device to the other. Telnet can accept either an IPv4 or IPv6 address or a domain name as the remote device address.

Prerequisites for Telnet

You have configured IP on a Layer 3 interface, out of band on the mgmt 0 interface, or inband on an Ethernet interface. Telnet supports both IPv4 and IPv6 addresses.

Guidelines and Limitations for Telnet

- The Telnet server is disabled by default
- Cisco NX-OS commands may differ from Cisco IOS commands.

Default Setting for Telnet

Parameter	Default
Telnet server	Enabled

Configuring Telnet

Enabling the Telnet Server

The Telnet server is enabled by default, but you can reenable it if necessary.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature telnet**	Enables the Telnet server.
Step 3	(Optional) switch(config)# **show telnet server**	Displays the Telnet server configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the Telnet server:

```
switch# configure terminal
switch(config)# feature telnet
switch(config)# show telnet server
telnet service enabled
switch(config)# copy running-config startup-config
```

Starting an IPv4 Telnet Session to a Remote Device

Before you begin

- Log in to the CLI in EXEC mode.
- Verify that the Telnet server is enabled and that it is also enabled on the remote device.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]	Creates an IP Telnet session to the specified destination. The keywords and arguments are as follows: • *port-number*—Port number, from 1 to 65535, to use for this session. The default port number is 23 • *vrf-name*—Default VRF.

Starting an IPv6 Telnet Session to a Remote Device

Before you begin

- Log in to the CLI in EXEC mode.
- Verify that the Telnet server is enabled and that it is also enabled on the remote device.
- Obtain the hostname for the remote device and, if needed, the username on the remote device.

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]	Creates an IP Telnet session to the specified destination. The keywords and arguments are as follows: • *port-number*—Port number, from 1 to 65535, to use for this session. The default port number is 23 • *vrf-name*—Default VRF.

Clearing Telnet Sessions

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user Telnet session.
Step 3	(Optional) switch# **show users**	Displays user session information.

Example

This example shows how to clear a Telnet session:

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/1        Jul 28 14:04   .         31453 (::ffff:171.70.209.8)
admin    pts/2        Jul 28 14:04   .         31475 (171.70.209.8)*
switch# clear line 1
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    tty1         Jul 25 19:13  old         2867
admin    pts/2        Jul 28 14:04   .         31475 (171.70.209.8)*
switch#
```

Verifying the Telnet Configuration

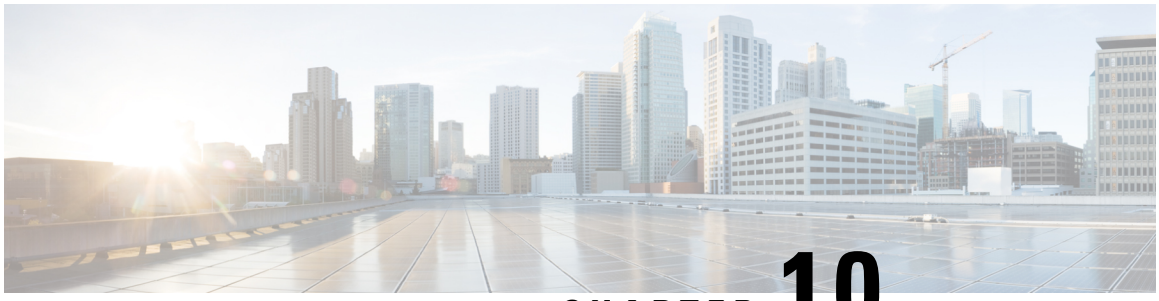
Use the following commands to verify the configuration.

Command	Purpose
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 telnet server	Displays the telnet server configuration.
show hosts	Displays the configuration details for current hosts.
show tcp connection	Displays connection information.

Feature History for Telnet

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name		Feature Information
IPv6 support	5.2(1)SV3(1.1)	IPv6 support was added.
Telnet	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 10

Configuring IP ACLs

This chapter contains the following sections:

Information About ACLs

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

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

ACL Types and Applications

An ACL is considered a port ACL when you apply it to one of the following:

- Ethernet interface
- vEthernet interface

When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on that trunk port. Both IPv4 and IPv6 ACLs are supported.

Active Ports and Services on Nexus 1000V VSM

The following table lists the active ports and services on Nexus 1000V VSM:

Port Number	Protocol	Remark
22	SSH / SCP /SFTP (TCP)	
23	TELNET (TCP)	
123	NTP (UDP)	Used by NTP Sever
161	SNMP (UDP)	Used by SNMP Server

Order of ACL Application

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

1. Ingress port ACL
2. Egress port ACL

Rules

Rules are what you create, modify, and remove when you configure how an access control list (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 all VEMs.

You can create rules in ACLs 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 to match the rule.

Source and Destination

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

How you specify the source and destination depends on whether you are configuring IP or MAC ACLs. For information about specifying the source and destination, see the applicable permit and deny commands in the *Cisco Nexus 1000V Command reference* .

Protocols

ACLs allow you to identify traffic by protocol. You can specify some protocols by name. For example, in an IP ACL, you can specify ICMP by name.

In IP 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.

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.

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 1000V Command Reference* .

Implicit Rules

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. Implicit rules ensure that unmatched traffic is denied, regardless of the protocol specified in the Layer 2 header of the traffic.

All IPv4 ACLs include the following implicit rule that denies unmatched IP traffic:

```
deny ip any any
```

All IPv6 ACLs include the following implicit rule:

```
permit icmp any any nd-na
permit icmp any any nd-ns
permit icmp any any router-advertisement
permit icmp any any router-solicitation
deny ipv6 any any
```

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

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

All MAC ACLs include the following implicit rule:

```
deny any any
```

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:

- IP 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

- MAC ACLs support the following additional filtering options:
  - Layer 3 protocol
  - VLAN ID
  - Class of Service (CoS)

See the *Cisco Nexus 1000V Command Reference* guide for information about filtering options available when using the applicable permit and deny commands.

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 by 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, you can 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.

Statistics

The device can maintain global statistics for each rule that you configure. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

Note The device does not support interface-level ACL statistics.

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

ACL Logging

You can use ACL logging to monitor flows that affect specific ACLs. The ACLs can be configured with the optional log keyword in each of the access control entries (ACEs). When you configure an option, statistics for each flow that match the ACL permit or deny conditions that you enter are logged in the software.

This example shows how to apply the log option to any IPv4 ACL:

```
switch(config)# ip access-list [name]
switch(config-acl)# permit tcp any 156.10.3.44/24 log
```

This example shows how to apply the log option to any IPv6 ACL:

```
switch(config)# ip access-list [name]
switch(config-acl)# permit tcp any 2001:0db8:85a3::/48 log
```

You can enable logging per rule(s) within the ACL. An implicit deny rule is the default action for ACLs. To log any packets that match the implicit deny rule, you must create an explicit deny rule and add the **log** keyword.

Note

- ACL logging is applicable only to IP ACLs that are configured with the **ip access-list** or **ipv6 access-list** commands. MAC ACL logging is not supported. Other traffic, such as the Virtual Supervisor Module (VSM) management interface or the selectors (aaa authen match, qos match, and so on), are not logged.
- ACL logging does not use the VSM management IP address. When in Layer 3 mode, ACL logging uses the Layer 3 vmk IP address. When in Layer 2 mode, ACL logging uses the vmk0 IP address.

Statistics and logging are provided for each flow. A flow is defined by the following IP flows:

- VSM ID
- Virtual Ethernet Module (VEM) ID
- Source interface
- Protocol
- Source IP address
- Source port
- Destination IP address

- Destination port

Scalability is provided through the following functionality:

- Each Cisco Nexus 1000V switch can support up to 256 VEMs.
- Each VEM can support up to 5000 permits and 5000 denies flows. The maximum number of permit/deny flows is a configurable option.
- The flow reporting interval can be set from 5 up to 86,400 seconds (1 day).
- The configuration flow syslog level can be from 0 to 7.
- Up to three syslog servers are supported.

ACL Flows

An ACL flow as it pertains to ACL logging has the following characteristics:

- It represents a stream of IPv4/IPv6 packets with the same packet headers (SrcIP, DstIP, Protocol, SrcPort, DstPort) for which an identical ACL action is enforced. Each flow entry tracks the count of packets that match the flow.
- It is created only if logging is enabled on the corresponding ingress/egress ACL policy. Ingress and egress flows are tracked separately.
- Each VEM tracks a maximum of 10,000 ACL flows; a flow space is shared between permit/deny flows, and each has a configurable maximum of 5000.
- Each flow entry contains the following:
  - Packet tuple
  - ACL action
  - Direction
  - Packet count
- The ACL flow life cycle is as follows:
  - A flow is created when the first packet of a unidirectional stream matches a Layer 3 ACL policy. A new flow notification is sent to the syslog server.
  - For all subsequent packets with a tuple that matches the flow tuple, the per flow packet counter is incremented.
  - Each flow is tracked periodically based on the configured reporting interval. Within each periodic report, all the active flows and the corresponding packet count seen since the last periodic report are reported to the syslog server
  - If no packets match a flow for one full periodic interval, the flow entry is purged. This process is the only flow-aging scheme.
  - A flow is not stateful. There is no connection tracking for TCP flows.
- The flow reporting process occurs in the following manner:

- For each flow created, a new flow notification message is sent to the syslog server.
- A periodic report for each active flow comes next. A flow is active if packets that match the flow are seen since the last periodic report.
- The flow information is exported to the syslog server and contains the following: packet tuple, ACL action, direction, VEM-ID, VSM-ID, packet count.
- The periodic time can be as low as 5 seconds with the default setting of 5 minutes. A new user space ACL-logging thread handles the periodic poll and report functionality.
- Syslog messages that identify the flow space usage are sent at 75 percent, 90 percent, and 100 percent of the threshold maximum to the syslog server once during each interval.

Syslog Messages

Syslog message characteristics are as follows:

- Syslog messages that contain flow information are exported from each Virtual Ethernet Module (VEM).
- The syslog client functionality is RFC-5424 compliant and communicates to servers over a UDP port (514).
- Any host that contains a VEM must be configured with a vmknic interface that can reach the remote syslog server.
- On an ESXi-5.0 host, syslog messages are blocked by a firewall. The Cisco Nexus 1000V has installation scripts that open the firewall for port 514.

Prerequisites for IP ACLs

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

Guidelines and Limitations for IP ACLs

ACLs are not supported in port channels.

Default Settings for IP ACLs

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

Configuring IP ACLs

Creating an IP ACL

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **{ip | ipv6} access-list** *name*	Creates the named IP ACL (up to 64 characters) and enters IP ACL configuration mode.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The sequence-number argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.
Step 4	(Optional) switch(config-acl)# **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch(config-acl)# **show** {**ip**|**ipv6**} **access-lists** *name*	Displays the IP ACL configuration.
Step 6	(Optional) switch(config-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
switch(config-acl)# copy running-config startup-config
```

This example shows how to create an IPv6 ACL:

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

Changing an IP ACL

You can add and remove rules in an existing IP ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and create it again with the desired changes.

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

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip**|**ipv6**} **access-list** *name*	Places you in IP ACL configuration mode for the specified ACL.
Step 3	(Optional) switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.
Step 4	(Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination}*	Removes the rule that you specified from the IP ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.
Step 5	(Optional) switch(config-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) switch(config-acl)# **show ip access-lists** *name*	Displays the IP ACL configuration.

	Command or Action	Purpose
Step 7	(Optional) switch(config-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration
Step 8	(Optional) switch(config-acl)# **exit**	Exit the ACL configuration mode for the new rules to take effect. **Note** If you are not executing this step, then wait for 5 seconds for the new rules to take effect.

Example

This example shows how to change an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
switch(config-acl)# ip access-list acl-01
switch(config-acl)# no 10
switch(config-acl)# no statistics per-entry
switch(config-acl)# show ip access-lists acl-01

IPV4 ACL acl-01
switch(config-acl)# copy running-config startup-config
```

Removing an IP ACL

Before you remove an IP ACL from the switch, ensure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty, that is, empty ACL with implicit rule of deny IP any .Use the **show ip access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

Before you begin

- Log in to the CLI in EXEC mode
- Know whether the ACL is applied to an interface.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **no** {**ip**|**ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	(Optional) switch(config)# **show** {**ip**|**ipv6**} **access-list** *name* **summary**	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to remove an IP ACL:

```
switch# configure terminal
switch(config)# no ip access-list acl-01
switch(config)# show ip access-lists acl-01 summary
switch(config)# copy running-config startup-config
```

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

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	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 from 1 to 4294967295.
Step 3	switch(config)# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change sequence numbers in an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
        20 permit ip 192.168.5.0/24 any
switch(config)# resequence ip access-list acl-01 100 10
switch(config)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any
        110 permit ip 192.168.5.0/24 any
switch(config)# copy running-config startup-config
```

Applying an IP ACL as a Port ACL

You can apply an IPv4 or IPv6 ACL to a physical Ethernet interface or a virtual Ethernet interface. ACLs applied to these interface types are considered port ACLs.

An IP ACL can also be applied on a port profile that is attached to a physical Ethernet interface or a virtual Ethernet interface.

Note ACLs cannot be applied on a port-channel interface. However, an ACL can be applied on a physical Ethernet interface that is not part of the port channel.

Before you begin

- Log in to the CLI in EXEC mode
- You can apply one port ACL to an interface.
- Check if the ACL that you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**vethernet** ¦ **ethernet**} *port*	Places you into interface configuration mode for the specified interface. **Note** Port ACLs are not supported on the port-channel interface and physical Ethernet interface that is a member of the port channel.

	Command or Action	Purpose
Step 3	switch(config-if)# {**ip port** | **ipv6 port**} {**access-group** | **traffic-filter**} *name* {**in** | **out**}	Adds the named IPv4 or IPv6 ACL to the port profile for either inbound or outbound traffic. You can apply only one IP port ACL to an interface.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

Example

This example shows how to apply an IP ACL as a port ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface vethernet 1
switch(config-if)# ip port access-group acl-01 in
switch(config-if)# show running-config aclmgr
ip access-list acl-01
  statistics per-entry
  100 permit ip 192.168.2.0/24 any
  110 permit ip 192.168.5.0/24 any
interface Vethernet1
  ip port access-group acl-01 in
switch(config-if)# copy running-config startup-config
```

Adding an IP ACL to a Port Profile

You can add an IPv4 or IPv6 ACL to a port profile.

You must know the following information:

- If you want to create a new port profile, you must know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- The name of the IP access control list that you want to configure for this port profile.
- The direction of the packet flow for the access list.

Before you begin

- Log in to the CLI in EXEC mode.
- Create the IP ACL to add to this port profile and you know its name.
- If you are using an existing port profile, you have created it and you know its name.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# {**ip port** | **ipv6 port**} {**access-group** | **traffic-filter**} *name* { **in** | **out**}	Adds the named IPv4 or IPv6 ACL to the port profile for either inbound or outbound traffic.
Step 4	(Optional) switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	Displays the configuration for verification.
Step 5	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add an IP ACL to a port profile:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile type vethernet vm_eth1
switch(config-port-prof)# ip port access-group acl-01 out
switch(config-port-prof)# show port-profile name vm_eth1
port-profile vm_eth1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  ip port access-group acl-01 out
  no shutdown
 evaluated config attributes:
  ip port access-group acl-01 out
  no shutdown
 assigned interfaces:
 port-group: vm_eth1
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static

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

Applying an IP ACL to the Management Interface

You can apply an IP ACL to the management interface, mgmt0.

Before you begin

Log in to the CLI in EXEC mode.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface mgmt0**	Places you into interface configuration mode for the management interface.
Step 3	switch(config-if)# {**ip**|**ipv6**} **access-group | traffic-filter** *access-list* [**in** | **out**]	Applies a specified inbound or outbound IP ACL to the interface.
Step 4	(Optional) switch(config-if)# **show** {**ip**|**ipv6**} **access-group | traffic-filter** *access-list*	Displays the ACL configuration.
Step 5	switch(config-if)# {**ip**|**ipv6**} **access-list match-local-traffic**	The **match-local-traffic** option enables matching for locally-generated traffic. **Note** This global command has to be enabled for ACL rules to take effect when the ACL is applied in the egress direction on the mgmt 0 interface.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to apply an IP ACL to the management interface:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit tcp any any
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        10 permit tcp any any
switch(config-acl)# interface mgmt 0
switch(config-if)# ip access-group acl-01 out
switch(config-if)# show ip access-lists acl-01 summary
IPV4 ACL acl-01
        Total ACEs Configured:1
        Configured on interfaces:
                mgmt0 -  egress (Router ACL)
        Active on interfaces:
                mgmt0 -  egress (Router ACL)
switch(config-if)# ip access-list match-local-traffic
switch(config)# copy running-config startup-config
```

Configuring ACL Logging

ACL logging is enabled by default on all Virtual Ethernet Modules (VEMs). In addition, the following guidelines apply to ACL logging configuration:

- Any rule can be enabled for logging by adding the **log** keyword.
- Only packets that have a rule with the **log** keyword enabled are logged.

Disabling ACL Logging

You can disable ACL logging on a VEM by entering the following command:

Command	Purpose
[**no**] **logging ip access-list cache module** *vem*	Disables ACL logging on the specified VEM.

Configuring a Time Interval for Accumulating Packet Counters

You can configure the time interval for accumulating packet counters before they are reported to the syslog servers. You enter the time range in seconds from 5 to 86,400 seconds (1 day). The default is 300 seconds (5 minutes).

You can configure the amount of time to accumulate packet counters by entering one of the following commands:

Command	Purpose
logging ip access-list cache interval *secs*	Sets the time interval in seconds to accumulate packet counters before they are reported to the syslog servers, where *secs* is the number of seconds.
[**no**] **logging ip access-list cache interval** *secs*	Reverts the configuration to the default time interval configuration 300 seconds (5 minutes), where *secs* is the number of seconds.

These examples show the time interval syslog message format that is sent periodically when the time interval expires:

```
ACL-LOGGING-6-PERMIT-FLOW-INTERVAL <VSM-id> <VEM-id> <protocol> <source-interface>
 <source-ip/source-port)> <destination-ip/destination-port> Hit-count = <nnn>

ACL-LOGGING-6-DENY-FLOW-INTERVAL <VSM-id> <VEM-id> <protocol> <source-interface>
<source-ip/source-port)> <destination-ip/destination-port> Hit-count = <nnn>
```

Configuring Flows

You can configure the number of deny and permit flows per VEM. The range is from 0 to 5000 flows. The default is 3000. A syslog message is sent when the flow is near the maximum threshold. The first message is sent when the number of flows has reached 75 percent of the maximum threshold and the next message is sent when the number of flows has reached 90 percent of the maximum threshold. The last message is sent when the number of flows reaches the maximum threshold of 100 percent.

Configuring Permit Flows

You can configure permit flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-permit-flows *num*	Sets the number of permit flows where *num* is the number of flows.
[**no**] **logging ip access-list cache max-permit-flows**	Reverts the configuration to the default permit flow value 3000.

These examples show permit flow syslog messages:

- New flow notification message:

```
-  Aug 28 04:17:19 fish-231-157.cisco.com 1 2011-08-28T11:14:23 - n1k-ecology -
ACLLOG-PERMIT-FLOW-CREATE VSM ID: 172.23.231.150, VEM ID:
86d04494-79e2-11df-a573-d0d0fd093c68, Source IP: 192.168.231.22, Destination IP:
192.168.231.21, Source Port: 42196, Destination Port: 8029, Source Interface: Veth2,
Protocol: "TCP"(6), Hit-count = 1
```

- Periodic flow reporting message:

```
-  Aug 28 04:17:20 sfish-231-157.cisco.com 1 2011-08-28T11:14:23 - n1k-acllog -
ACLLOG-PERMIT-FLOW-INTERVAL VSM ID: 172.23.231.150, VEM ID:
86d04494-79e2-11df-a573-d0d0fd093c68, Source IP: 192.168.231.22, Destination IP:
192.168.231.21, Source Port: 42196, Destination Port: 8029, Source Interface: Veth2,
Protocol: "TCP"(6), Hit-count = 1245
```

- Threshold crossing alarm messages:

```
-  Aug 28 04:17:22 sfish-231-157.cisco.com 1 2011-08-28T11:14:24 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 75 percent

limit (3969)
-  Aug 28 04:17:26 sfish-231-157.cisco.com 1 2011-08-28T11:14:26 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 90 percent

limit (4969)
-  Aug 28 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Configuring Deny Flows

You can configure deny flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-deny-flows *num*	Sets the number of deny flows, where *num* is the number of flows.
[**no**] **logging ip access-list cache max-deny-flows**	Reverts the configuration back to the default deny flow value 3000.

These examples show deny flow syslog messages:

- New flow notification message

```
-  Aug 28 04:17:19 sfish-231-157.cisco.com 1 2011-08-28T11:14:23 - n1k-acllog -
ACLLOG-DENY-FLOW-CREATE VSM ID: 172.23.231.150, VEM ID:
86d04494-79e2-11df-a573-d0d0fd093c68, Source IP: 192.168.231.22, Destination IP:
192.168.231.100, Source Port: 48528, Destination Port: 8029, Source Interface: Veth2,
Protocol: "TCP"(6), Hit-count = 1
```

- Periodic flow reporting message

```
-  Aug 28 04:17:20 sfish-231-157.cisco.com 1 2011-08-28T11:14:23 - n1k-acllog -
ACLLOG-DENY-FLOW-INTERVAL VSM ID: 172.23.231.150, VEM ID:
86d04494-79e2-11df-a573-d0d0fd093c68, Source IP: 192.168.231.22, Destination IP:
192.168.231.100, Source Port: 47164, Destination Port: 8029, Source Interface: Veth2,
Protocol: "TCP"(6), Hit-count = 1245
```

- Threshold crossing alarm messages

```
-  Aug 28 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 75 percent
limit
(4330)
-  Aug 28 04:18:27 sfish-231-157.cisco.com 1 2011-08-28T11:15:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 90 percent
limit
(4630)
-  Aug 28 04:20:17 sfish-231-157.cisco.com 1 2011-08-28T11:17:20 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Syslog Server Severity Levels

You can configure severity levels of the ACL logging syslog messages for up to three remote syslog servers. The range is from 0 to 7. The default severity level is 6.

Severity Code	Severity Level	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant condition
6	Informational	informational messages
7	Debug	Debug-level messages

Setting the Severity Level for a Syslog Message

You can set the severity level of a syslog message and the server to which you want the message to be sent by entering one of the following commands:

Command	Purpose
[**no**] **acllog match-log-level** *level*	Sets the severity level at which syslog messages are sent, where *level* is the severity code from 0 to 7. The no acllog match-log-level level command will revert the ACL log level back to the default severity level 6.
[**no**] **logging ip access-list cache max-deny-flows** *number*	Sets the maximum number of deny flows to *number* per module. The **no logging ip access-list cache max-deny-flows** *number* sets the maximum number of deny-flows to default value of 3000.
[**no**] **logging ip access-list cache max-permit-flows** *number*	Set the max-permit-flows to a specified number per module. The **no logging ip access-list cache max-permit-flows** *number* sets the maximum number of permit-flows to default value of 3000.
logging server { *A.B.C.D* | *x:x:x:x:x:x:x:x*}*-level*	Specifies the syslog server on which you want to set a severity level, where *A.B.C.D* is the syslog server IPv4 address and *x:x:x:x:x:x:x:x* is the syslog server IPv6 address. The severity levels are between 0 to 7.

Note For ACL logging to work, ACL Logging level should be equal or less than that of Syslog level.

Verifying the IP ACL Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including the IP ACL configuration and interfaces that IP ACLs are applied to.
show {**ip**|**ipv6**} **access-lists** [*name*]	Displays all IP ACLs or a named IP ACL.
show {**ip**|**ipv6**} **access-lists** [*name*] su**mmary**	Displays a summary of all configured IP ACLs or a named IP ACL.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.
show logging ip access-list status	Displays the ACL logging configuration for a VSM.
vemcmd show acllog config	Displays the VEM ACL logging configuration.

Monitoring IP ACLs

Use one of the following commands for IP ACL monitoring:

Command	Purpose
show {**ip**|**ipv6**} **access-lists**	Displays the IPv4 or IPv6 ACL configuration. If the IP ACL includes the **statistics per-entry** command, the output includes the number of packets that have matched each rule.
clear {**ip**|**ipv6**} **access-list** [*name*] **counters**	Clears statistics for all IPv4 or IPv6 ACLs or for a specific IPv4 or IPv6 ACL.

Configuration Example for IP ACL

This example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL on physical ethernet interface which is not a member of port-channel and configuration verification with match counters:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# permit ip 192.168.5.0/24 any
switch(config-acl)# permit 22 any 10.105.225.225/27
switch(config-acl)# permit ip any 10.105.225.225/27
switch(config-acl)# statistics per-entry
switch(config-acl)# interface ethernet 3/5
switch(config-if)# ip port access-group acl-01 in
switch(config-if)# show ip access-lists acl-01 summary
IPV4 ACL acl-01
        statistics per-entry
        Total ACEs Configured:4
        Configured on interfaces:
                Ethernet3/5 - ingress (Port ACL)
        Active on interfaces:
                Ethernet3/5 - ingress (Port ACL)
switch(config-if)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any [match=0]
        110 permit ip 192.168.5.0/24 any [match=0]
        120 permit 22 any 10.105.225.225/27 [match=0]
        130 permit ip any 10.105.225.225/27 [match=44]
switch(config-if)# clear ip access-list counters acl-01
switch(config-if)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any [match=0]
        110 permit ip 192.168.5.0/24 any [match=0]
        120 permit 22 any 10.105.225.225/27 [match=0]
        130 permit ip any 10.105.225.225/27 [match=0]
switch(config-if)#
```

This example shows how to enable access list matching for locally generated traffic:

```
switch# ip access-list match-local-traffic
```

This example shows how to verify VSM ACL logging configuration:

```
switch# show logging ip access-list status
Max deny flows = 3000
Max permit flows = 3000
Alert interval = 300
Match log level = 6
VSM IP = 192.168.1.1
Syslog IP = 10.1.1.1
Syslog IP = 0.0.0.0
Syslog IP = 0.0.0.0
ACL Logging enabled on module(s):
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
ACL Logging disabled on module(s):
3
```

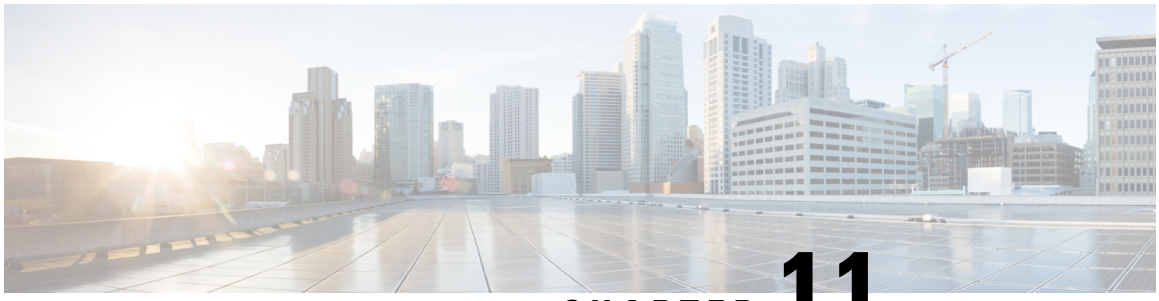
This example shows how to verify VEM ACL logging configuration:

```
switch# vemcmd show acllog config
ACL-Log Config:
Status: enabled
Reporting Interval: 300
Max Permit Flows: 3000
Max Deny Flows: 3000
Syslog Facility : 4
Syslog Severity: 6
Syslog Srvr 1: 10.1.1.1
Syslog Srvr 2: 0.0.0.0
Syslog Srvr 3: 0.0.0.0
VSM: 192.168.1.1
```

Feature History for IP ACLs

This table only includes updates for those releases that have resulted in additions to the feature.

Feature History	Releases	Feature Information
IPv6 ACLs	5.2(1)SV3(1.1)	This feature was introduced.
IPv6 ACL Logging	5.2(1)SV3(1.1)	
IPv4 ACL Logging	4.2(1)SV1(5.1)	This feature was introduced.
IP ACLs for mgmt0 interface	4.2(1) SV1(4)	This feature was introduced.
IP ACLs	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 11

Configuring MAC ACLs

This chapter contains the following sections:

Information About MAC ACLs

MAC access control lists (ACLs) are ACLs that filter traffic using information in the Layer 2 header of each packet.

Prerequisites for MAC ACLs

- You must be familiar with MAC addressing and non-IP protocols to configure MAC ACLs.
- You must be familiar with the ACL concepts presented in this document.

Guidelines and Limitations for MAC ACLs

ACLs are not supported in port channels.

Default Settings for MAC ACLs

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

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it. You can also use this procedure to add the ACL to a port profile.

Before you begin

- Log in to the CLI in EXEC mode.
- Have a name to assign to the ACL that you are creating.
- Create a port profile if you want to add the ACL to it.

If you want to also add the ACL to a port profile, you must know the following:

- If you are using an existing port profile, you have already created it and you know its name.
- The interface type (Ethernet or vEthernet) and the name that you want to give the port profile if you are creating a new port profile.
- The direction of packet flow for the access list.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.

	Command or Action	Purpose
Step 4	(Optional) switch(config-mac-acl)# **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 6	(Optional) switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# copy running-config startup-config
```

Changing a MAC ACL

You can change an existing MAC ACL, for example, to add or remove rules.

Use the **resequence** command to reassign sequence numbers, such as when adding rules between existing sequence numbers.

Before you begin

- Log in to the CLI in EXEC mode.
- In an existing MAC ACL, know that you cannot change existing rules.
- In an existing MAC ACL, know that you can add and remove rules.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	(Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*	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

	Command or Action	Purpose
		sequence number, the rule is added to the end of the rules. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.
Step 4	(Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}	Removes the rule that you specify from the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference*.
Step 5	switch(config-mac-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 7	switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show mac access-lists

MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# no 10
switch(config-mac-acl)# no statistics per-entry
switch(config-mac-acl)# end
switch# show mac access-lists

MAC ACL acl-mac-01
        20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch# copy running-config startup-config
```

Removing a MAC ACL

You can remove a MAC ACL from the switch. Ensure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where the ACL is applied. Instead, the switch considers the removed ACL to be empty.

To find the interfaces that a MAC ACL is configured on, use the **show mac access-lists** command with the summary keyword.

Before you begin

- Log in to the CLI in EXEC mode.
- Know whether the ACL is applied to an interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no mac access-list** *name*	Removes the specified MAC ACL from the running configuration.
Step 3	(Optional) switch(config)# **show mac access-lists** *name* **summary**	Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to remove a MAC ACL:

```
switch# configure terminal
switch(config)# no mac access-list acl-mac-01
switch(config)# show mac access-lists acl-mac-01 summary
switch(config)# copy running-config startup-config
```

Changing Sequence Numbers in a MAC ACL

You can change 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.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence mac access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change sequence numbers in a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
        20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# resequence mac access-list acl-mac-01 100 10
switch(config)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        100 permit 00c0.4f00.0000 0000.00ff.ffff any
        110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# copy running-config startup-config
```

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:

- Physical Ethernet interfaces
- Virtual Ethernet interfaces

A MAC ACL can also be applied to a port profile that is attached to a physical Ethernet interface or a virtual Ethernet interface.

Note ACLs cannot be applied on a port-channel interface. However, an ACL can be applied on a physical Ethernet interface that is not part of the port channel.

Before you begin

- Log in to the CLI in EXEC mode.
- Know that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** { **ethernet** | **vethernet** } *port*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# **mac port access-group** *access-list* [**in** | **out**]	Applies a MAC ACL to the interface.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to apply a MAC ACL as a port ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface vethernet 1
switch(config-if)# mac port access-group acl-mac-01 in
switch(config-if)# show running-config aclmgr
mac access-list acl-mac-01
  100 permit 00C0.4F00.0000 0000.00FF.FFFF any
  110 permit F866.F222.E5A6 FFFF.FFFF.FFFF any
interface Vethernet1
  mac port access-group acl-mac-01 in
switch(config-if)# copy running-config startup-config
```

Adding a MAC ACL to a Port Profile

You can add a MAC ACL to a port profile.

Before you begin

- Log in to the CLI in EXEC mode.
- Create the MAC ACL to add to this port profile and know its name.
- If you are using an existing port profile, know its name.
- If you are creating a new port profile, know the interface type (Ethernet or vEthernet) and the name you want to give the profile.

- Know the direction of packet flow for the access list.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Places you in port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **mac port access-group** *name* {**in** | **out**}	Adds the named ACL to the port profile for either inbound or outbound traffic.
Step 4	(Optional) switch(config-port-prof)# **show port-profile name** *profile-name*	Displays the configuration for verification.
Step 5	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add a MAC ACL to a port profile:

```
switch# configure termnal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile vm_eth1
switch(config-port-prof)# mac port access-group acl-mac-01 out
switch(config-port-prof)# show port-profile name vm_eth1
port-profile vm_eth1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  mac port access-group acl-mac-01 out
  no shutdown
 evaluated config attributes:
  mac port access-group acl-mac-01 out
  no shutdown
 assigned interfaces:
 port-group: vm_eth1
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static

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

Verifying MAC ACL Configurations

Use one of the following commands to verify the configuration:

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 they are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL.
show mac access-lists summary	Displays a summary of all configured MAC ACLs or a named MAC ACLs.

Monitoring MAC ACLs

Use the following commands for MAC ACL monitoring:

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.

Configuration Examples for MAC ACLs

Configuration Example for Creating a MAC ACL for any Protocol

This example shows how to create an MAC ACL named acl-mac-01 and apply it as a port ACL on physical ethernet interface which is not a member of port-channel and configuration verification with match counters.

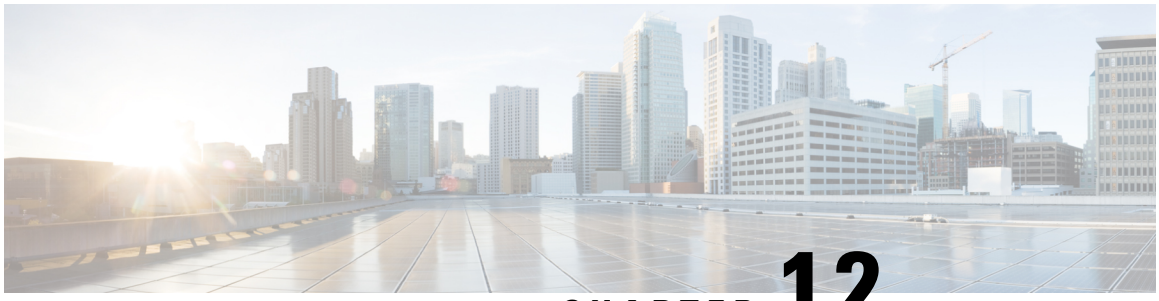
```
switch(config)# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# 100 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# 110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# end
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 3/5
switch(config-if)# mac port access-group acl-mac-01 out
```

```
switch(config-if)# show mac access-lists acl-mac-01 summary

MAC ACL acl-mac-01
        statistics per-entry
        Total ACEs Configured:2
        Configured on interfaces:
                Ethernet3/5 -  egress (Port ACL)
        Active on interfaces:
                Ethernet3/5 -  egress (Port ACL)
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
        110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=546]
switch(config-if)# clear mac access-list counters
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
        110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=0]
switch(config-if)#
```

Feature History for MAC ACLs

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
MAC ACL	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 12

Configuring Port Security

This chapter contains the following sections:

Information About Port Security

Port security allows you to configure Layer 2 interfaces that permit inbound traffic from a restricted, secured set of MAC addresses. Traffic from secured MAC addresses is not allowed on another interface within the same VLAN. The number of MAC addresses that can be secured is configured per interface.

Secure MAC Address Learning

The following information describes secure MAC address learning:

- The process of securing a MAC address is called learning.
- The number of addresses that can be learned is restricted.
- Address learning can be accomplished on any interface where port security is enabled.

Static Method

- The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are persistent if the device restarts.
- A static secure MAC address entry remains in the configuration of an interface until you explicitly remove the address from the configuration.

- Adding secure addresses by the static method is not affected by whether dynamic or sticky address learning is enabled.
- The burned-in MAC address is secured as a static MAC address starting from Release 5.2(1)SV3(1.1). In previous releases, the burned-in MAC address was secured as a dynamic MAC address.

Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- The VSM and VEM restarts.
- The interface restarts.
- The address reaches the age limit that you configured for the interface.
- You explicitly remove the address.

The burned-in MAC address is secured as a static MAC address starting from Release 5.2(1)SV3(1.1). In previous releases, the burned-in MAC address was secured as a dynamic MAC address.

Sticky Method

- If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning. These addresses can be made persistent through a reboot by using the **copy run start** command to copy the running configuration to the startup configuration.
- Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, dynamic learning is stopped and sticky learning is used instead. If you disable sticky learning, dynamic learning is resumed.
- Sticky secure MAC addresses are not aged.
- A sticky secure MAC address entry remains in the configuration of an interface until you explicitly remove the address.

Dynamic Address Aging

MAC addresses that are learned by the dynamic method are aged and dropped when reaching the age limit. You can configure the age limit on each interface. The range is from 0 to 1440 minutes, where 0 disables aging.

There are two methods of determining the address age:

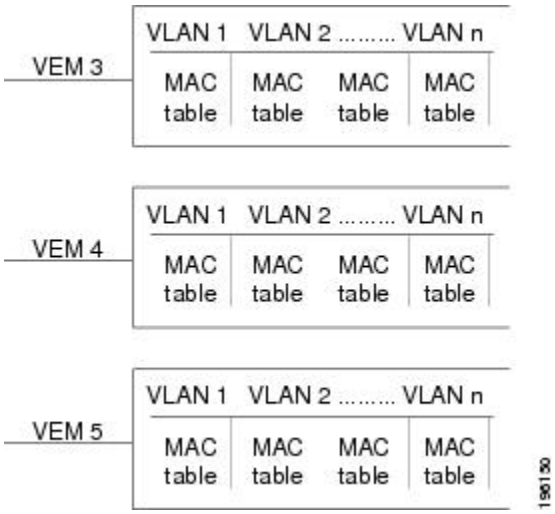
- Inactivity—The length of time after the device last received a packet from the address on the applicable interface.
- Absolute—The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Secure MAC Address Maximums

The secure MAC addresses on a secure port are inserted in the same MAC address table as other regular MAC addresses. If a MAC table has reached its limit, it does not learn any new secure MAC addresses for that VLAN.

The following figure shows that each VLAN in a VEM has a forwarding table that can store a maximum number of secure MAC addresses.

Figure 8: Secure MAC Addresses per VEM



Interface Secure MAC Addresses

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.

The following limits can determine how many secure MAC address are permitted on an interface:

- Device maximum— 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 for both access and trunk vethernet ports. Interface maximums cannot exceed the device maximum.
- VLAN maximum—You can configure the maximum number 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.

The maximum number of secure MAC addresses per port is limited to ten. When configuring ports in trunk mode, be sure not to exceed the maximum MAC address limit.

You can configure a VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first.

Security Violations and Actions

Port security triggers a security violation when the following occurs:

Note Beginning with Release 5.2(1)SV3(1.1), MAC move detection and violation is local to a VEM.

- Ingress traffic arrives at an interface from a nonsecure MAC address and learning the address would exceed the applicable maximum number of secure MAC addresses.

  When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

  - VLAN 1 has a maximum of five addresses.
  - The interface has a maximum of ten addresses.

  A violation is detected when either of the following occurs:

  - Five addresses are learned for VLAN 1 and inbound traffic from a sixth address arrives at the interface in VLAN 1.
  - Ten addresses are learned on the interface and inbound traffic from an 11th address arrives at the interface.

When a security violation occurs on an interface, the action specified in its port security configuration is applied. 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.

```
switch(config)# errdisable recovery cause psecure-violation
switch(config)# copy running-config startup-config
```

- 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.
- Restrict—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 and causes the security violation counter to increment.

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 veth 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 Ports—Port security is not supported on Ethernet ports.
- Ethernet Port Channels—Port security is not supported on Ethernet port channels.

Result of Changing an Access Port to a 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.

Result of Changing a Trunk Port to an 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.

Beginning with Release 5.2(1)SV3(1.1), the maximum number of secure MAC addresses per port is limited to 10. When configuring ports in trunk mode, be sure not to exceed the maximum MAC address limit. If you configure an interface in trunk mode that exceeds the MAC address limit and you attempt to change the mode to access, the interface might be left with stale secure MAC address entries.

Guidelines and Limitations for Port Security

- Port security is not supported on the following:
  - Ethernet interfaces
  - Ethernet port-channel interfaces
  - Switched port analyzer (SPAN) destination ports
- Port security cannot be configured on interfaces with existing static MAC addresses.
- Port security cannot be enabled on interfaces whose VLANs have an existing static MAC address even if it is programmed on a different interface.
- If the interface maximum has been reached for secure MAC addresses and you add an additional static MAC address, the interface enters error-disable mode. To enable the interface, you must first remove the static MAC address using the **no switchport port-security mac-address** command and then use the **shutdown** and **no shutdown** commands on the interface. To avoid this issue, before adding additional static MAC addresses, use the **show port-security address interface** *veth-number* command to verify whether the interface maximum has been reached.

- Beginning with Release 5.2(1)SV3(1.1), the maximum number of secure MAC addresses per port is limited to 10. When configuring ports in trunk mode, be sure not to exceed the maximum MAC address limit. If you configure an interface in trunk mode that exceeds the MAC address limit and you attempt to change the mode to access, the interface might be left with stale secure MAC address entries.

Default Settings for Port Security

Parameters	Default
Interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown

Configuring Port Security

Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface.

By default, port security is disabled on all interfaces.

Enabling port security on an interface also enables dynamic MAC address learning.

Before you begin

- Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security**	Enables port security on the interface. Using the **no** option disables port security on the interface.
Step 4	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.

	Command or Action	Purpose
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable port security on a Layer 2 interface:

```
switch# configure terminal
switch(config)# interface vethernet 36
switch(config-if)# switchport port-security
switch(config-if)# show running-config port-security
interface Vethernet36
switchport port-security
switch(config-if)# show port-security address interface vethernet 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0050.5687.3C68 DYNAMIC Vethernet36 0
------------------------------------------------------------------------
switch(config-if)# show port-security interface vethernet 36
Port Security : Enabled
Port Status : Secure UP
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Security violation count : 0

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

Enabling or Disabling Sticky MAC Address Learning

You can enable or disable sticky MAC address learning.

Dynamic MAC address learning is the default on an interface.

By default, sticky MAC address learning is disabled.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security mac-address sticky**	Enables sticky MAC address learning on the interface. Using the **no** option disables sticky MAC address learning.
Step 4	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable sticky MAC address learning:

```
switch(config)# interface Vethernet36
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security mac-address sticky
switch(config-if)# switchport port-security mac-address 0050.5687.3C4B
switch(config)# show running-config port-security
interface Vethernet36
switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address 0050.5687.3C4B
switch(config)# show port-security address interface vethernet 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2304 0050.5687.3C4B STICKY Vethernet36 0
------------------------------------------------------------------------
```

Adding a Static Secure MAC Address on an Interface

You can add a static secure MAC address on an interface.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

- Log in to the CLI in EXEC mode.
- Determine if the interface maximum has been reached for secure MAC addresses. If the interface maximum has been reached for secure MAC addresses and you add an additional static MAC address, the interface enters error-disable mode. To enable the interface, you must first remove the static MAC address using the **no switchport port-security mac-address** command and then use the **shutdown** and **no shutdown** commands on the interface. To avoid this issue, before adding additional static MAC addresses, use the **show port-security address interface** *veth-number* command to verify whether the interface maximum has been reached.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]	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	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add a static secure MAC address on an interface:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 2
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security mac-address 0019.0002.0102
switch(config-if)# show running-config port-security

!Command: show running-config port-security
!Time: Tue Aug 12 23:49:23 2014

version 5.2(1)SV3(1.1)
```

```
interface Vethernet2
switchport port-security
switchport port-security maximum 5
switchport port-security mac-address 0019.0002.0102

switch(config-if)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
51 0019.0002.0102 STATIC Vethernet2 0
switch(config-if)# copy running-config startup-config
```

Removing a Static or a Sticky Secure MAC Address from an Interface

Starting in Release #5.2(1)SV3(1.1), the Sticky MAC address is stored only on the Virtual Ethernet Module (VEM) and not on the Virtual Supervisor Module (VSM). The stored MAC addresses that are secured using Sticky MAC address configuration do not persist across events such as **vMotions**, **Port Group Change**, and **Interface Disconnect from VC**.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# **no switchport port-security mac-address** *address*	Removes the MAC address from port security on the current interface.
Step 4	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to remove the MAC address from port security on the current interface:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 2
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security mac-address 0019.0002.0102
switch(config-if)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
51 0019.0002.0102 STATIC Vethernet2 0
switch(config-if)# no switchport port-security mac-address 0019.0002.0102
switch(config-if)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
switch(config-if)# copy running-config startup-config
```

Removing a Dynamic Secure MAC Address

You can remove a specific address learned by the dynamic method or remove all addresses learned by the dynamic method on a specific interface.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **clear port-security dynamic** {**interface vethernet** *number* | **address** *address* | **module** *module-numbef*} [**vlan** *vlan-ID*]	Removes dynamically learned, secure MAC addresses, as specified. The keywords are as follows: • **interface**—Removes all dynamically learned addresses on the interface that you specify. • **address**—Removes the single, dynamically learned address that you specify. • **module**—Removes all dynamically learned addresses on the specified module.

	Command or Action	Purpose
		• **vlan**—Removes an address or addresses on a particular VLAN.
Step 3	(Optional) switch(config)# **show port-security address**	Displays secure MAC addresses.

Example

This example shows how to remove a dynamically learned, secure MAC address by specifying the vethernet number:

```
switch(config)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
51 0010.0201.0500 DYNAMIC Vethernet2 0
switch(config)# clear port-security dynamic interface vethernet 2
switch(config)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
switch(config)#
```

This example shows how to remove a dynamically learned, secure MAC address by specifying the module number:

```
switch(config)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
51 0010.0201.0500 DYNAMIC Vethernet2 0

switch(config)# clear port-security dynamic address 0010.0201.0500 module 3
switch(config)# show port-security address interface vethernet 2
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.1f1c STATIC Vethernet2 0
switch(config)#
```

Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface.

The secure MAC addresses share the Layer 2 Forwarding Table (L2FT). The forwarding table for each VLAN can hold up to 10 entries.

By default, an interface has a maximum of one secure MAC address.

VLANs have no default maximum number of secure MAC addresses.

To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

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 command is rejected.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]	Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. 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. **Note** The maximum number of MAC addresses that can be secured on an interface is ten. However, the command allows you to configure 1,025. We recommend that you do not configure more than ten.
Step 4	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.

	Command or Action	Purpose
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN ID configuration is not supported on access port and is only applicable to trunk ports.

Example

This example shows how to configure a maximum number of MAC addresses:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 2
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security maximum 5
switch(config-if)# show port-security interface vethernet 2
Port Security : Enabled
Violation Mode : Shutdown
Aging Time : 0
Aging Type : Absolute
Maximum MAC Addresses : 5
Total MAC Addresses : 1
Configured MAC Addresses : 1
Sticky MAC Addresses : 0
Security violation count : 0
switch(config-if)# show running-config port-security

!Command: show running-config port-security
!Time: Wed Aug 13 00:56:49 2014

version 5.2(1)SV3(1.1)

interface Vethernet2
switchport port-security
switchport port-security maximum 5

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

Configuring an Address Aging Type and Time

You can configure the MAC address aging type and the length of time used to determine when MAC addresses learned by the dynamic method have reached their age limit.

There are two methods for determining address aging:

- 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.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security aging type** {**absolute** | **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	switch(config-if)# [**no**] **switchport port-security aging time** *minutes*	Configures the number of minutes that a dynamically learned MAC address must age before the address is dropped. The maximum valid minutes is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).
Step 5	(Optional) switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.
Step 6	(Optional) switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 7	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 8	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure an address aging type and time:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 3
switch(config-if)# switchport port-security
```

```
switch(config-if)# switchport port-security aging type inactivity
switch(config-if)# switchport port-security aging time 120
switch(config-if)# show port-security address interface vethernet 3
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.38ec STATIC Vethernet3 0
51 0000.0000.0010 DYNAMIC Vethernet3 120
switch(config-if)# show port-security interface vethernet 3
Port Security : Enabled
Violation Mode : Shutdown
Aging Time : 120
Aging Type : Inactivity
Maximum MAC Addresses : 5
Total MAC Addresses : 2
Configured MAC Addresses : 1
Sticky MAC Addresses : 0
Security violation count : 0
switch(config-if)# copy running-config startup-config
switch(config-if)# show running-config port-security

!Command: show running-config port-security
!Time: Wed Aug 13 01:06:00 2014

version 5.2(1)SV3(1.1)

interface Vethernet3
switchport port-security
switchport port-security aging type inactivity
switchport port-security aging time 120
switchport port-security maximum 5
```

Configuring a Security Violation Action

You can configure how an interface responds to a security violation. You can configure the following interface responses to security violations:

- **protect**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value.
- **restrict**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value and causes the SecurityViolation counter to increment.
- **shutdown** (the default)—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable port security on the interface that you are configuring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# [**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**}	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. The keywords are as follows: • **protect**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value • **restrict**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value, which increments the Security Violation counter. • **shutdown** (the default)—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification and syslog event.
Step 4	switch(config-if)# **show port-security address interface vethernet number**	Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	Displays the port security configuration on the interface.
Step 6	(Optional) switch(config-if)# **show running-config port-security**	Displays the port security configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a security violation action:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 3
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security violation protect
```

```
switch(config-if)# show port-security interface vethernet 3
Port Security : Enabled
Violation Mode : Protect
Aging Time : 120
Aging Type : Inactivity
Maximum MAC Addresses : 5
Total MAC Addresses : 2
Configured MAC Addresses : 1
Sticky MAC Addresses : 0
Security violation count : 0
# copy running-config startup-config

switch(config-if)# show running-config port-security

!Command: show running-config port-security
!Time: Wed Aug 13 01:14:41 2014

version 5.2(1)SV3(1.1)

interface Vethernet3
switchport port-security
switchport port-security aging type inactivity
switchport port-security aging time 120
switchport port-security maximum 5
switchport port-security violation protect
```

Recovering Ports Disabled for Port Security Violations

You can automatically recover an interface disabled for port security violations. To recover an interface manually from the error-disabled state, you must enter the **shutdown** command and then the **no shutdown** command.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery cause psecure-violation**	Enables a timed automatic recovery of the specified port that is disabled for a port security violation.
Step 3	switch(config)# **errdisable recovery interval** *seconds*	Configures a timer recovery interval in seconds from 30 to 65535 seconds.

Example

This example shows how to recover ports that are disabled for port security violations:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# errdisable recovery cause psecure-violation
switch(config)# errdisable recovery interval 30
switch(config)# copy running-config startup-config
switch(config)# show errdisable recovery
ErrDisable Reason Timer Status
----------------- ------------
link-flap disabled
bpduguard disabled
dhcp-rate-limit enabled
arp-inspection enabled
security-violation disabled
psecure-violation enabled
failed-port-state enabled
ip-addr-conflict disabled

Timer interval: 30
```

Verifying the Port Security Configuration

Use the following commands to verify the configuration:

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security address interface vethernet number	Displays the secure MAC address learnt on the interface.
show port-security interface vethernet number	Displays the port security configuration on the interface.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses.

Configuration Example for Port Security

This example shows a port security configuration for the vEthernet 3 interface with a VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to protect.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 3
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security maximum 10
switch(config-if)# switchport port-security maximum 6 vlan 50
switch(config-if)# switchport port-security maximum 3 vlan 55
switch(config-if)# switchport port-security violation protect
```

```
switch(config-if)# switchport mode trunk
switch(config-if)# show running-config interface vethernet 3
interface Vethernet3
switchport mode trunk
switchport port-security
switchport port-security maximum 10
switchport port-security violation protect
switchport port-security maximum 6 vlan 50
switchport port-security maximum 3 vlan 55
switch(config)# copy running-config startup-config
```

The following example shows a port security configuration for the vEthernet 3 interface as an access port with an interface maximum set to 10, a violation set to restrict, an absolute timeout of 1 minute and a port security static MAC address of 0000.1111.5555:

```
switch(config)# interface vethernet 3
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security aging time 1
switch(config-if)# switchport port-security aging type absolute
switch(config-if)# switchport port-security maximum 10
switch(config-if)# switchport port-security mac-address 0000.1111.5555
switch(config-if)# switchport port-security violation restrict
switch(config-if)# show running-config interface vethernet 3
interface Vethernet3
switchport port-security
switchport port-security aging time 1
switchport port-security maximum 10
switchport port-security violation restrict
switchport port-security mac-address 0000.1111.5555
switchport port-security aging type absolute
no shutdown

switch(config-if)# show port-security interface vethernet 3
Port Security : Enabled
Violation Mode : Restrict
Aging Time : 1
Aging Type : Absolute
Maximum MAC Addresses : 10
Total MAC Addresses : 7
Configured MAC Addresses : 7
Sticky MAC Addresses : 0
Security violation count : 0
switch(config-if)# copy running-config startup-config
```

This example shows a port security configuration for the vEthernet 3 interface as an access port with a violation set to shutdown, maximum count to 2 and MAC address learning set to sticky:

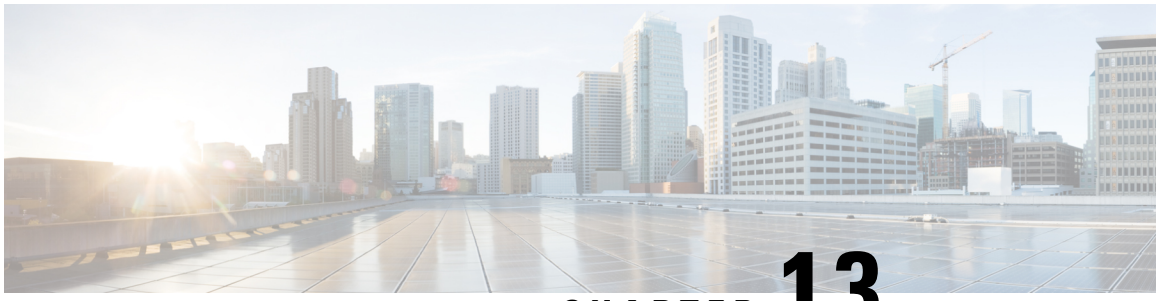
```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface vethernet 3
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security mac-address sticky
switch(config-if)# switchport port-security violation shutdown
switch(config-if)# switchport port-security maximum 2
switch(config-if)# show running-config interface vethernet 3
interface Vethernet3
inherit port-profile 51
switchport port-security violation shutdown
switchport port-security
switchport port-security maximum 2
switchport port-security mac-address sticky
switch(config-if)# show port-security interface vethernet 3
Port Security : Enabled
```

```
Violation Mode : Shutdown
Aging Time : 0
Aging Type : Absolute
Maximum MAC Addresses : 2
Total MAC Addresses : 2
Configured MAC Addresses : 1
Sticky MAC Addresses : 1
Security violation count : 0
switch(config-if)# show port-security address interface vethernet 3
Secure Mac Address Table
------------------------------------------------------------------------
Vlan/Vxlan Mac Address Type Ports Configured Age
(mins)
---------- ----------- ------ ----- ---------------
51 0050.56a4.38ec STATIC Vethernet3 0
51 0000.0000.0010 STICKY Vethernet3 0
switch(config-if)# copy running-config startup-config
```

Feature History for Port Security

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
MAC address per port limit	5.2(1)SV3(1.1)	The maximum number of secure MAC addresses per port is limited to ten.
MAC Move Detection and Violation	5.2(1)SV3(1.1)	This feature is now local to VEM.
Port Security	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 13

Configuring DHCP Snooping

This chapter contains the following sections:

Information About DHCP Snooping

DHCP snooping functions like a firewall between untrusted hosts and trusted DHCP servers by doing the following:

- Validates DHCP messages received from untrusted sources and filters out invalid response messages from DHCP servers.
- 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 globally and per VLAN. By default, DHCP snooping is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

DHCP Overview

The Dynamic Host Configuration Protocol (DHCP) provides the configuration parameters to Internet hosts. DHCP does the following:

- Delivers host-specific configuration parameters from a DHCP server to a host.
- Allocates network addresses to hosts.

DHCP is built on a client/server model, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts.

By default, DHCP supports the following mechanisms for IP address allocation:

- Automatic allocation—- DHCP assigns a permanent IP address to a client.
- Dynamic allocation—DHCP assigns an IP address to a client for a limited period of time (or until the client explicitly relinquishes the address).
- Manual allocation—The network administrator assigns an IP address to a client and DHCP is used to convey the assigned address to the client.

The format of DHCP messages is based on the format of Bootstrap Protocol (BOOTP) messages. This format supports BOOTP relay agent functionality and interoperability between BOOTP clients and DHCP servers. With BOOTP relay agents, you do not need to deploy a DHCP server on each physical network segment.

DHCP uses the two ports assigned by IANA for BOOTP. The destination UDP port 67 sends data to the server, and UDP port 68 sends data to the client.

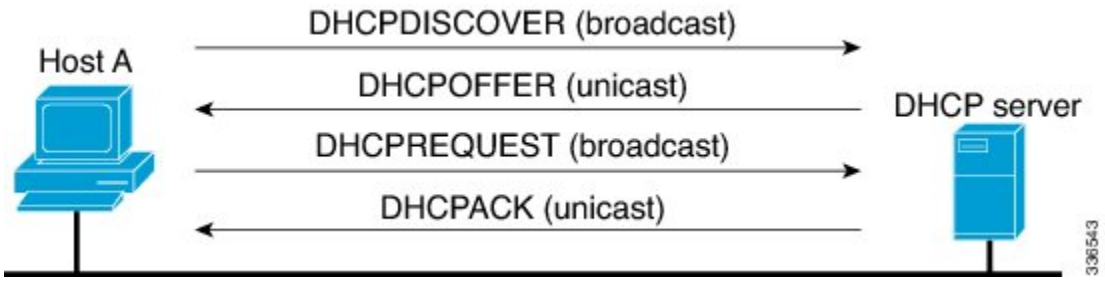
DHCP operations are categorized into four basic phases:

- IP Discovery
- IP Lease Offer
- IP Request
- IP Lease Acknowledgement

Note The DHCP operations phases are often abbreviated as DORA (Discovery, Offer, Request, and Acknowledgement).

The following figure shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a Cisco IOS DHCP server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast message.

Figure 9: DHCP Request for an IP Address from a DHCP Server



The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client.

BOOTP Packet Format

BOOTP requests and replies are encapsulated in UDP datagrams as shown in the following figure and table.

Figure 10: BOOTP Packet Format

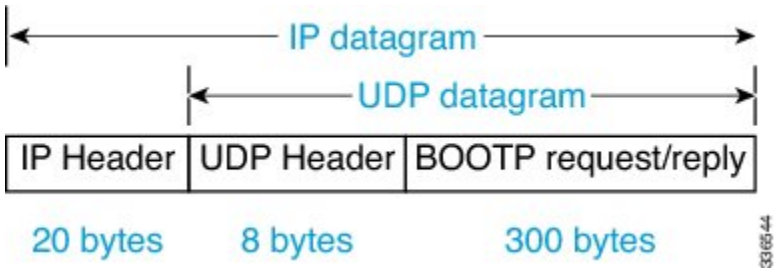


Figure 11: 300-Byte BOOTP Request and Reply Format

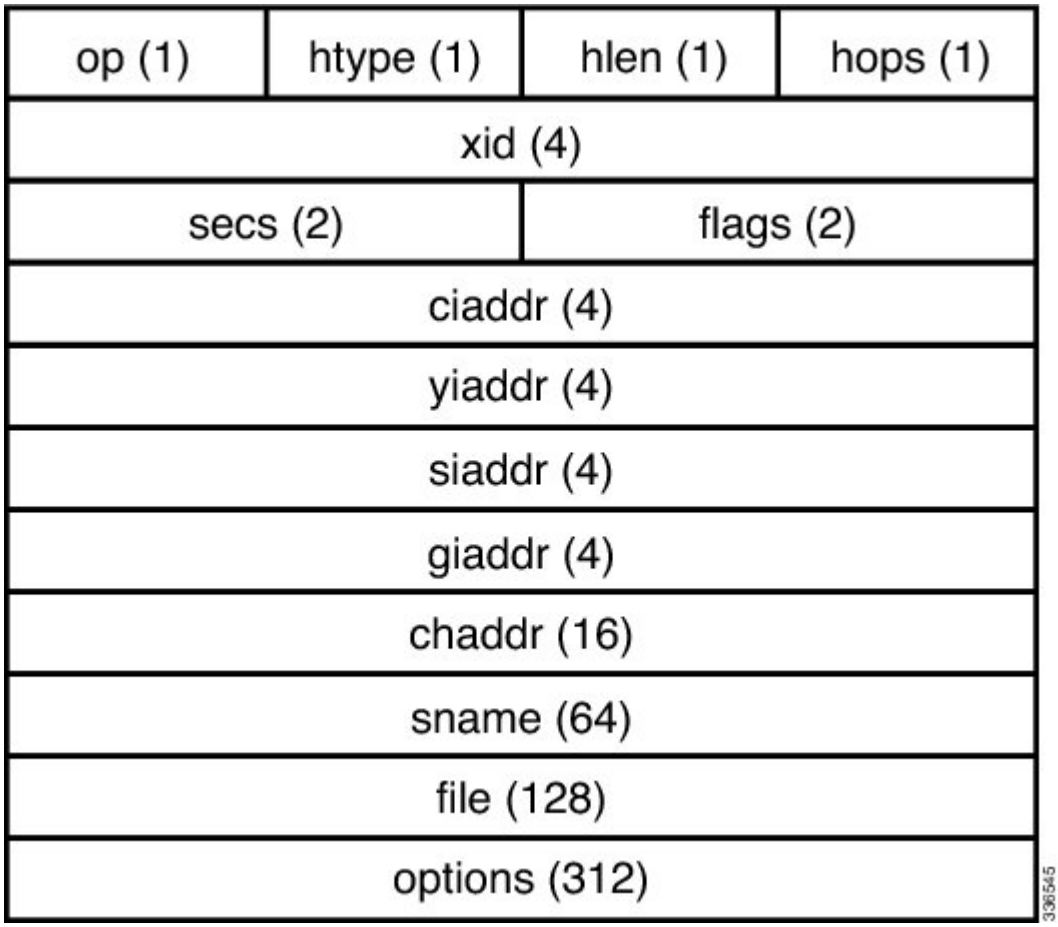


Table 5: BOOTP Request and Reply Format

Field	Bytes	Name	Description
op	1	OpCode	Identifies the packet as a request or reply. 1=BOOTREQUEST and 2=BOOTREPLY.
htype	1	Hardware Type	Specifies the network hardware type.
hlen	1	Hardware Length	Specifies the length hardware address length.
hops	1	Hops	The client sets the value to zero and the value increments if the request is forwarded across a router.

Field	Bytes	Name	Description
xid	4	Transaction ID	A random number that is chosen by the client. All DHCP messages exchanged for a given DHCP transaction use the ID (xid).
secs	2	Seconds	Specifies number of seconds since the DHCP process started.
flags	2	Flags	Indicates whether the message will be broadcast or unicast.
ciaddr	4	Client IP Address	Used when the client is aware of the IP address as in the case of the Bound, Renew, or Rebinding states.
yiaddr	4	Your IP Address	If the client IP address is 0.0.0.0, the DHCP server places the offered client IP address in this field.
siaddr	4	Server IP Address	If the client knows the IP address of the DHCP server, this field is populated with the DHCP server address. Otherwise, it is used in DHCPOFFER and DHCPACK from the DHCP server.
giaddr	4	Router IP Address	The gateway IP address, filled in by the DHCP/BootP Relay Agent.
chaddr	16	Client MAC Address	The DHCP client MAC address.
sname	64	Server Name	The optional server hostname.
File	128	Boot Filename	The boot filename.

Field	Bytes	Name	Description
Options	Variable	Option Parameters	The optional parameters that can be provided by the DHCP server. RFC 2132 lists all possible options.

Trusted and Untrusted Sources

DHCP snooping identifies ports as trusted or untrusted sources. When you enable DHCP snooping, by default, all vEthernet (vEth) ports are untrusted and all Ethernet ports (uplinks), port channels, special vEth ports (used by other features, such as the Virtual Service Domain (VSD) are trusted.

In an enterprise network, a trusted source is a device that is under your administrator's control. Any device beyond the firewall or outside the network is an untrusted source. Client ports are generally treated as untrusted sources.

In the Cisco Nexus 1000V switch, you indicate that a source is trusted by configuring the trust state of its connecting interface. Uplink ports, as defined with the uplink capability on port profiles, are trusted and cannot be configured to be untrusted.

DHCP snooping does the following and acts like a firewall between untrusted clients and trusted DHCP servers:

- Only DHCP messages that come from a server that is connected to a trusted port are accepted. Any DHCP message on UDP port 68 that is data from the server to the client that is received on an untrusted port is dropped.
- Builds and maintains the DHCP snooping binding database, which contains information about clients with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from clients.

By default, DHCP snooping is inactive on all VLANs. You can enable DHCP snooping on a single VLAN or a range of VLANs. DHCP snooping is enabled globally and per VLAN.

DHCP Snooping Binding Database

By using the information that is extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database on each Virtual Ethernet Module (VEM). The database contains an entry for each untrusted client with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts that are connected through trusted interfaces.

Note The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

DHCP snooping updates the database when the device receives specific DHCP messages. For example, with DHCP snooping, you can add an entry to the database when the device receives a DHCPACK message from the server. DHCP snooping also allows you to remove an entry in the database when the IP address lease

expires or the device receives a DHCPRELEASE or DHCP DECLINE from the DHCP client or a DHCPNACK from the DHCP server.

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.

To remove dynamically added entries from the binding database, use the **clear ip dhcp snooping binding** command.

DHCP Snooping Option 82 Data Insertion

DHCP can centrally manage the IP address assignments for a large number of subscribers. When you enable option 82, the device identifies a subscriber device that connects to the network using the vEthernet number to which the client is connected and the Virtual Supervisor Module (VSM) to which the client belongs (in addition to its MAC address). Multiple hosts on the subscriber LAN can connect to the same port on the access device and are uniquely identified.

When you enable option 82 on the Cisco Nexus 1000V, the following sequence of events is displayed:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Cisco Nexus 1000V Virtual Ethernet Module (VEM) 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), the port identifier, and the vEth number from which the packet is received (the circuit ID suboption).
3. The device forwards the DHCP request that includes the option 82 field to the DHCP server.
4. The DHCP server receives the packet. If the server is option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the option 82 field in the DHCP reply.
5. The DHCP server sends the reply to the Cisco Nexus 1000V. The Cisco Nexus 1000V verifies that it originally inserted the option 82 data by inspecting the remote ID and the circuit ID fields. The Cisco Nexus 1000V VEM removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

Option 82 Insertion

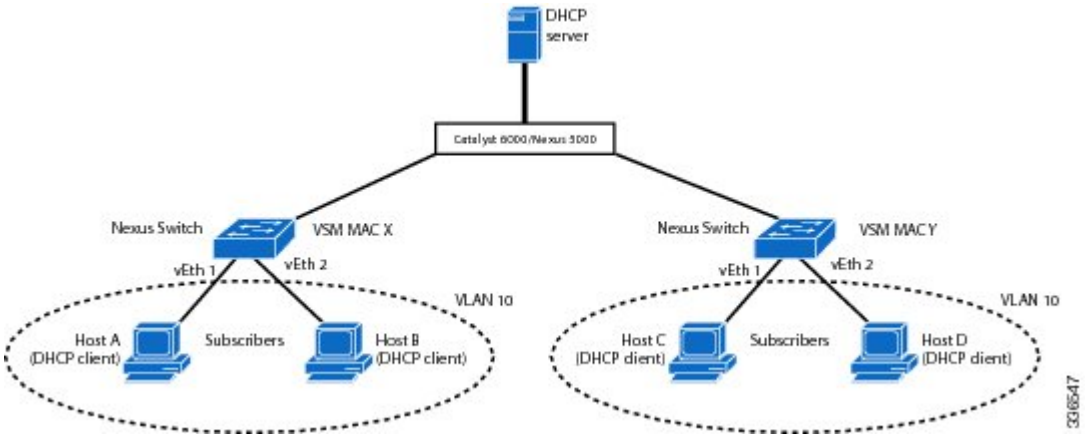
The following figure describes a typical use case of option 82 insertion. Host A and Host B are part of Cisco Nexus 1000V with the VSM MAC address X on VLAN 10. Similarly, Host C and Host D are part of the Cisco Nexus 1000 V with the VSM MAC address Y also on VLAN 10. All the clients receive an IP address from the common DHCP server that is connected to the upstream switch.

Option 82 insertion enables you to assign specific IP addresses to Host C and Host A. These hosts are both part of VLAN 10 and have the same vEth numbers (vEthernet1). You can also assign IP addresses to Hosts D and Host B (vEthernet 2) by using the VSM MAC address in the DHCP packet.

DHCP packets from clients A and B hosted on the first Cisco Nexus 1000V have the VSM MAC X in the Remote ID field whereas requests from clients C and D have the VSM MAC Y in the Remote ID field. Based on the remote IDs, you can configure the DHCP server with pools to assign separate

set of IPs to clients on each Cisco Nexus 1000V even though the clients are part of the same VLAN (VLAN 10).

Figure 12: Option 82 Insertion Topology



Suboption Packet Formats

The following figure shows the packet formats for the remote ID suboption and the circuit ID suboption. The Cisco Nexus 1000V uses these packet formats when you globally enable DHCP snooping and when you enable option 82 data insertion and removal. For the circuit ID suboption, the circuit ID string is the name of the vEth port to which the client is connected. For the Remote ID suboption, the MAC address is the Asynchronous Inter-process Communication (AIPC) interface on the Cisco Nexus 1000V.

Figure 13: Remote ID Suboption Frame Format

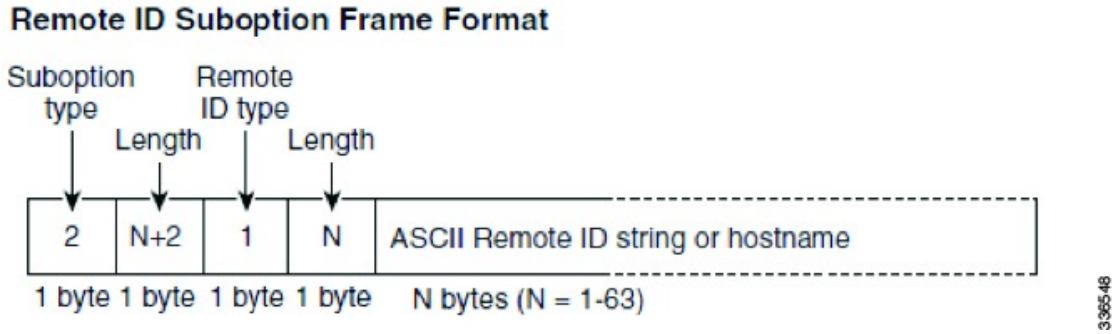
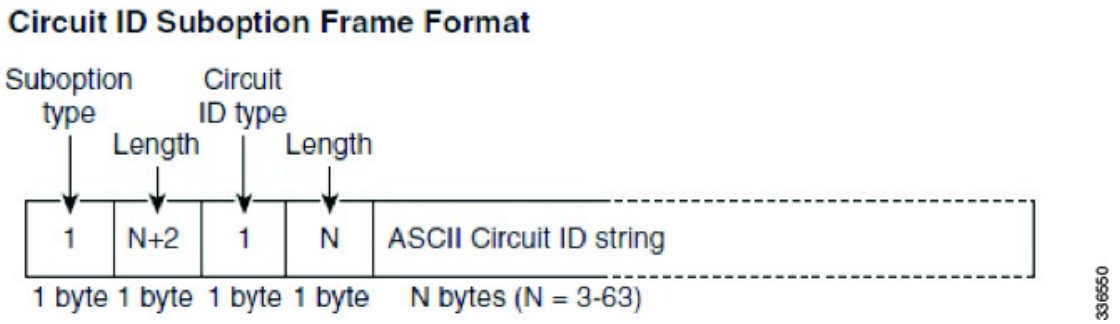


Figure 14: Circuit ID Suboption Frame Format



Licensing Requirements for DHCP Snooping

The following table shows the licensing requirements for this feature:

Feature	License Requirement
DHCP snooping	Starting with Release 4.2(1)SV2(1.1), a tier-based licensing approach is adopted for the Cisco Nexus 1000V. The Cisco Nexus 1000V is shipped in two editions: Essential and Advanced. When the switch edition is configured as the Advanced edition, DHCP snooping, Dynamic ARP Inspection (DAI), and IP Source Guard (IPSG) are available as advanced features that require licenses. **Note** Starting with Release 4.2(1)SV2(1.1), you can enable DHCP snooping on the Cisco Nexus 1000V by using the **feature dhcp** command. If the switch edition is Essential, the **feature** command fails. See the *Cisco Nexus 1000V License Configuration Guide* for more information about the licensing requirements for the Cisco Nexus 1000V.

Prerequisites for DHCP Snooping

- You must be familiar with DHCP to configure DHCP snooping.
- See the Licensing Requirements section for information about the licensing requirements of this feature.

Guidelines and Limitations for DHCP Snooping

- For seamless DHCP snooping, Virtual Service Domain (VSD) service VM ports are trusted ports by default. If you configure these ports as untrusted, this setting is ignored.
- If the VSM uses the VEM for connectivity (that is, the VSM has its VSM Asynchronous Inter-process Communication (AIPC), management, and inband ports on a particular VEM), you must configure these virtual Ethernet interfaces as trusted interfaces.
- You must configure connecting interfaces on a device upstream from the Cisco Nexus 1000V as trusted if DHCP snooping is enabled on the device.
- Enabling DHCP snooping on the primary VLAN enables snooping on all its corresponding secondary VLANs. Enabling DHCP snooping only on a secondary VLAN is not a valid configuration.
- You cannot enable DHCP snooping on VXLAN ports.

Default Settings for DHCP Settings

Parameters	Default
DHCP feature	Disabled
DHCP snooping global	Disabled
DHCP snooping VLAN	Disabled
DHCP snooping MAC address verification	Enabled
DHCP snooping trust	Trusted for Ethernet interfaces, vEthernet interfaces, and port channels in the VSD feature. Untrusted for vEthernet interfaces not participating in the VSD feature.
DHCP snooping limit rate	None

Configuring DHCP Snooping

Process for DHCP Snooping Configuration

1. Enable the DHCP feature.
2. Enable DHCP snooping globally.
3. Enable DHCP snooping on at least one VLAN.

   By default, DHCP snooping is disabled on all VLANs.
4. Ensure that the DHCP server is connected to the device using a trusted interface.

Enabling or Disabling the DHCP Feature

By default, DHCP is disabled.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DHCP snooping globally. The **no** option disables DHCP snooping but saves an existing DHCP snooping configuration. DHCP snooping is available as an advanced feature that requires a license. See the *Cisco Nexus 1000V License Configuration Guide* for more information about the licensing requirements for the Cisco Nexus 1000V.
Step 3	(Optional) switch(config)# **show feature**	Displays the state (enabled or disabled) of each available feature.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable DHCP:

```
switch# configure terminal
switch(config)# feature dhcp
switch(config)# show feature
Feature Name          Instance  State
--------------------  --------  --------
dhcp-snooping         1         enabled
http-server           1         enabled
lacp                  1         enabled
netflow               1         disabled
port-profile-roles    1         enabled
private-vlan          1         disabled
sshServer             1         enabled
tacacs                1         enabled
telnetServer          1         enabled
switch(config)# copy running-config startup-config
```

Enabling or Disabling DHCP Snooping Globally

Be sure you know the following information about DHCP snooping:

- By default, DHCP snooping is globally disabled.
- If DHCP snooping is globally disabled, all DHCP snooping stops and no DHCP messages are relayed.
- If you configure DHCP snooping and then globally disable it, the remaining configuration is preserved.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DHCP globally. DHCP snooping is available as an advanced feature that requires a license.
Step 3	switch(config)# [**no**] **ip dhcp snooping**	Enables IP DHCP snooping. The **no** option disables DHCP snooping but saves an existing DHCP snooping configuration.
Step 4	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 5	(Optional) switch(config)# **show ip dhcp snooping**	Displays the DHCP snooping IP configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable or disable DHCP snooping globally:

```
switch# configure terminal
switch(config)# feature dhcp
switch(config)# ip dhcp snooping
switch(config)# show running-conifg dhcp
feature dhcp
ip dhcp snooping
no ip dhcp relay
switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:none
DHCP snooping is operational on the following VLANs:none
Insertion of Option 82 is disabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 No Unlimited
switch(config)# copy running-config startup-config
```

Enabling or Disabling DHCP Snooping on a VLAN

By default, DHCP snooping is disabled on all VLANs.

Note Enabling DHCP snooping on the primary VLAN enables snooping on all its corresponding secondary VLANs. Enabling DHCP snooping only on a secondary VLAN is not a valid configuration.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DHCP globally. DHCP snooping is available as an advanced feature that requires a license.
Step 3	switch(config)# [**no**] **ip dhcp snooping vlan vlan-list**	Enables DHCP snooping on the VLANs specified by the VLAN-list. The **no** option disables DHCP snooping on the VLANs specified.
Step 4	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 5	(Optional) switch(config)# **show ip dhcp snooping**	Displays the DHCP snooping IP configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

Note Ensure the VLANs on which DHCP snooping is enabled are operational. If DHCP snooping is not operational on a VLAN, check if the VLAN is configured on Cisco Nexus 1000V and is in the active state.

This example shows how to enable or disable DHCP snooping on a VLAN:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip dhcp snooping vlan 100,200,250-252
switch(config)# show running-config dhcp
feature dhcp
```

```
ip dhcp snooping
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:100,200,250-252
DHCP snooping is operational on the following VLANs:100,200,250-252
Insertion of Option 82 is disabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 No Unlimited
switch(config)# copy running-config startup-config
```

Enabling or Disabling DHCP Snooping for MAC Address Verification

You can enable or disable DHCP snooping for MAC address verification. If the device receives a packet on an untrusted interface and the source MAC address and the DHCP client hardware address do not match, address verification causes the device to drop the packet. MAC address verification is enabled by default.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ip dhcp snooping verify mac-address**	Enables the DHCP snooping for MAC address verification. The **no** option disables MAC address verification.
Step 3	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 4	(Optional) switch(config)# **show ip dhcp snooping**	Displays the DHCP snooping IP configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable DHCP snooping for MAC address verification:

```
switch# configure terminal
switch(config)# ip dhcp snooping verify mac-address
switch(config)# show running-config dhcp
feature dhcp
```

```
ip dhcp snooping
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:100,200,250-252
DHCP snooping is operational on the following VLANs:100,200,250-252
Insertion of Option 82 is disabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 No Unlimited
switch(config)# copy running-config startup-config
```

Configuring an Interface as Trusted or Untrusted

You can configure whether a virtual Ethernet (vEth) interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust using one of the following methods:

- Layer 2 vEthernet interfaces
- Port profiles for Layer 2 vEthernet interfaces

By default, vEth interfaces are untrusted. The only exception is the special vEth ports that are used by other features, such as Virtual Service Domain (VSD), are trusted.

For seamless DHCP snooping, Dynamic ARP Inspection (DAI), IP Source Guard, VSD service VM ports are trusted ports by default. If you configure these ports as untrusted, this setting is ignored.

Before you begin

- Log in to the CLI in EXEC mode.
- Know that the vEthernet interface is configured as a Layer 2 interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified vEthernet interface. Use this command to configure an interface as a trusted interface using an interface configuration.
Step 3	switch(config)# **port-profile** *profilename*	Places you in port profile configuration mode for the specified port profile. Configures an interface as a trusted interface using a port profile configuration.

	Command or Action	Purpose
Step 4	switch(config-if)# [**no**] **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 5	(Optional) switch(config-if)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an interface as trusted or untrusted:

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# ip dhcp snooping trust
switch(config)# port-profile vm-data
switch(config-port-prof)# ip dhcp snooping trust
switch(config-port-prof)# show running-config dhcp
feature dhcp
interface Vethernet3
ip dhcp snooping trust
ip dhcp snooping
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config-port-prof)# copy running-config startup-config
```

Configuring the Rate Limit for DHCP Packets

You can configure a limit for the rate of DHCP packets per second received on each port.

Before you begin

Log in to the CLI in EXEC mode.

You should know the following information:

- Ports are put into an errdisabled state if they exceed the limit you set in this procedure for the rate of DHCP packets per second.
- You can configure the rate limit on either the interface or port profile.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified vEthernet interface.
Step 3	switch(config)# **port-profile** *profilename*	Places you in port profile configuration mode for the specified port profile.
Step 4	switch(config-if)# [**no**] **ip dhcp snooping limit rate** *rate*	Configures the limit for the rate of DHCP packets per second (1 to 2048). The **no** option removes the rate limit.
Step 5	(Optional) switch(config-if)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a rate limit for DHCP packets:

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# ip dhcp snooping limit rate 15
switch(config-if)# show running-config dhcp
feature dhcp
interface Vethernet3
ip dhcp snooping trust
ip dhcp snooping limit rate 15
ip dhcp snooping
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config-if)# copy running-config startup-config

switch(config)# port-profile vm-data
switch(config-port-prof)# ip dhcp snooping limit rate 15
switch(config-port-prof)# show running-config dhcp
feature dhcp
interface Vethernet3
ip dhcp snooping trust
ip dhcp snooping limit rate 15
ip dhcp snooping
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:100,200,250-252
DHCP snooping is operational on the following VLANs:100,200,250-252
Insertion of Option 82 is disabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 Yes 15
```

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

Detecting Disabled Ports for DHCP Rate Limit Violations

You can globally detect the disabled ports that exceed the DHCP rate limit.

To recover an interface manually from the error-disabled state, you must enter the **shutdown** command and then the **no shutdown** command.

Note A failure to conform to the set rate causes the port to be put into an errdisable state.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DHCP globally. DHCP snooping is available as an advanced feature that requires a license.
Step 3	switch(config)# [**no**] **errdisable detect cause dhcp-rate-limit**	Enables DHCP error-disabled detection. The **no** option disables DHCP error-disabled detection.
Step 4	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 5	(Optional) switch(config)# **show errdisable detect**	Displays the reasons for the port to be in the error-disabled state.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to detect disabled ports for DHCP rate limit violation:

```
switch# configure terminal
switch(config)# errdisable detect cause dhcp-rate-limit
switch(config)# show errdisable detect
ErrDisable Reason Timer Status
----------------- ------------
link-flap enabled
bpduguard enabled
```

```
dhcp-rate-limit enabled
arp-inspection enabled
ip-addr-conflict enabled
switch(config)# copy running-config startup-config
```

Recovering Disabled Ports for DHCP Rate Limit Violations

You can globally configure the automatic recovery of disabled ports for violating the DHCP rate limit.

To recover an interface manually from the error-disabled state, you must enter the **shutdown** command and then the **no shutdown** command.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **errdisable recovery cause dhcp-rate-limit**	Enables DHCP error-disabled detection. The **no** option disables DHCP error-disabled detection.
Step 3	switch(config)# **errdisable recovery interval** *time interval*	Sets the DHCP error-disabled recovery interval, where *time interval* is the number of seconds from 30 to 65535.
Step 4	(Optional) switch(config)# **show errdisable recovery**	Displays the recovery interval for the vEth to recover from the error-disabled state.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to recover disabled ports for DHCP rate limit violations:

```
switch# configure terminal
switch(config)# errdisable detect cause dhcp-rate-limit
switch(config)# show errdisable detect
ErrDisable Reason Timer Status
----------------- ------------
link-flap enabled
bpduguard enabled
dhcp-rate-limit enabled
arp-inspection enabled
ip-addr-conflict enabled
switch(config)# copy running-config startup-config
```

Clearing the DHCP Snooping Binding Database

You can clear the DHCP snooping binding database.

Clearing All Binding Entries

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear ip dhcp snooping binding**	Clears dynamically added entries from the DHCP snooping binding database.
Step 2	(Optional) switch# **show ip dhcp snooping binding**	Displays the DHCP snooping binding database.

Example

This example shows how to clear all binding entries:

```
switch# clear ip dhcp snooping binding
switch# show ip dhcp snooping binding
```

Clearing Binding Entries for an Interface

Before you begin

- Log in to the CLI in EXEC mode
- Collect the following information for the interface:
  - VLAN ID
  - IP address
  - MAC address

Procedure

	Command or Action	Purpose
Step 1	switch# **clear ip dhcp snooping binding** [{**vlan** *vlan-id* **mac** *mac-addr* **ip** *ip-addr* **interface** *interface-id*} | **vlan** *vlan-id1* | **interface** *interface-id1*]	Clears dynamically added entries for an interface from the DHCP snooping binding database.
Step 2	switch# **show ip dhcp snooping binding**	Displays the DHCP snooping binding database.

Example

This example shows how to clear binding entries for an interface:

```
switch# clear ip dhcp snooping binding vlan 10 mac EEEE.EEEE.EEEE ip 10.10.10.1 interface
vethernet 1
switch# show ip dhcp snooping binding
```

Relaying Switch and Circuit Information in DHCP

You can globally relay the VSM MAC address and vEthernet port information in DHCP packets.

Before you begin

Log in to the CLI in EXEC mode.

Note In a HA pair setup, the MAC address inserted in the option 82 field of the DHCP packet is the AIPC interface of the current active VSM. The match criteria on the DHCP server must match the AIPC MAC address of both primary and secondary VSMs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ip dhcp snooping information option**	Configures DHCP to relay the VSM MAC address and vEthernet port information in DHCP packets. Use the **no** option to remove this configuration.
Step 3	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP snooping running configuration.
Step 4	(Optional) switch(config)# **show ip dhcp snooping**	Displays the DHCP snooping IP configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to relay switch and circuit information in DHCP:

```
switch# configure terminal
switch(config)# ip dhcp snooping information option
switch(config)# show running-config dhcp
feature dhcp
```

```
interface Vethernet3
ip dhcp snooping trust
ip dhcp snooping limit rate 15
ip dhcp snooping
ip dhcp snooping information option
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:100,200,250-252
DHCP snooping is operational on the following VLANs:100,200,250-252
Insertion of Option 82 is enabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 Yes 15
switch(config)# copy running-config startup-config
```

Adding or Removing a Static IP Entry

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ip source binding** *ip address MAC address* **vlan** *vlanid* **interface vethernet** *interface-number*	Creates a static IP source entry for the current interface. Use the **no** option to remove the static IP source entry.
Step 3	(Optional) switch(config)# **show ip dhcp snooping binding interface vethernet** *interface number*	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	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add or remove a static IP entry:

```
switch# configure terminal
switch(config)#  ip source binding 10.5.22.178 001f.28bd.0014 vlan 100 interface vethernet
 3
switch(config)# show ip dhcp snooping binding interface vethernet 3
MacAddress          IpAddress        LeaseSec  Type        VLAN  Interface
------------------  ---------------  --------  ----------  ----  -------------
00:1f:28:bd:00:14  10.5.22.178      infinite   static      100   Vethernet3
switch(config)# copy running-config startup-config
```

Verifying the DHCP Snooping Configuration

Use the following commands to verify the configuration:

Command	Purpose
show running-config dhcp	Displays the DHCP snooping configuration.
show ip dhcp snooping	Displays general information about DHCP snooping.
show ip dhcp snooping binding	Displays the contents of the DHCP snooping binding table.
show feature	Displays the features available, such as DHCP, and whether they are enabled.

Monitoring DHCP Snooping

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping statistics.

```
switch(config)# show ip dhcp snooping statistics

Packets processed 0
Packets forwarded 0
Total packets dropped 0
Packets dropped from untrusted ports 0
Packets dropped due to MAC address check failure 0
Packets dropped due to Option 82 insertion failure 0
Packets dropped due to o/p intf unknown 0
Packets dropped which were unknown 0
Packets dropped due to service dhcp not enabled 0
Packets dropped due to no binding entry 0
Packets dropped due to interface error/no interface 0
Packets dropped due to max hops exceeded 0
```

Configuration Example for DHCP Snooping

This example shows how to enable DHCP snooping on VLAN 100, with vEthernet interface 5 trusted because the DHCP server is connected to that interface. This example shows how to configure a rate limit of 15 pps on the interface where the client is connected. The clients are using port-profile client-pp. When the rate limit is violated, the client port is put in the error-disabled state for 60 seconds before it is recovered. One of the clients has static DHCP IP assigned and one IP address has an infinite lease time assigned by the DHCP server:

```
switch# configure terminal
switch(config)# feature dhcp
switch(config)# ip dhcp snooping
switch(config)# ip dhcp snooping vlan 100
switch(config)# interface vethernet 3
switch(config-if)# ip dhcp snooping trust

switch(config)# port-profile type vethernet client-pp
switch(config-port-prof)# ip dhcp snooping limit rate 15
```

```
switch(config-port-prof)# errdisable detect cause dhcp-rate-limit
switch(config)# errdisable recovery interval 60
switch(config)# ip source binding 192.0.2.1 001f.28bd.0014 vlan 100 interface vethernet 3
switch(config)# show feature
Feature Name Instance State
---------------------------- -------- --------
bgp 1 disabled
cts 1 disabled
dhcp 1 enabled
dot1x 1 disabled
evb 1 disabled
http-server 1 enabled
lacp 1 disabled
netflow 1 disabled
network-segmentation-manager 1 disabled
port-profile-roles 1 disabled
private-vlan 1 disabled
privilege 1 disabled
scheduler 1 disabled
scp-server 1 disabled
segmentation 1 disabled
sftp-server 1 disabled
ssh 1 enabled
tacacs+ 1 disabled
telnet 1 disabled
vff 1 disabled
vtracker 1 disabled
vxlan-gateway 1 disabled
switch(config)# show running-config dhcp
feature dhcp
interface Vethernet3
ip dhcp snooping trust
ip dhcp snooping limit rate 15
ip dhcp snooping
ip dhcp snooping information option
ip dhcp snooping vlan 100,200,250-252
no ip dhcp relay
ip source binding 192.0.2.1 001f.28bd.0014 vlan 100 interface Vethernet3

switch(config)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:
100,200,250-252
DHCP snooping is operational on the following VLANs:
100,200,250-252
Insertion of Option 82 is enabled
Verification of MAC address is enabled
DHCP snooping trust is configured on the following interfaces:
Interface Trusted Pkt Limit
------------ ------- ---------
Vethernet1 No Unlimited
Vethernet2 No Unlimited
Vethernet3 Yes 15
switch(config)# copy running-config startup-config
switch(config)# show ip dhcp snooping binding
MacAddress IpAddress LeaseSec Type VLAN Interface
------------------ --------------- -------- ---------- ---- -------------
00:1f:28:bd:00:14 10.5.22.178 infinite static 100 Vethernet3
```

Note An entry with an infinite lease time issued by the DHCP server has infinite in the Lease Sec column and will be of Type dhcp-snoop.

When client interfaces are part of a secondary VLAN, the DHCP binding table displays the entries on its corresponding primary VLAN.

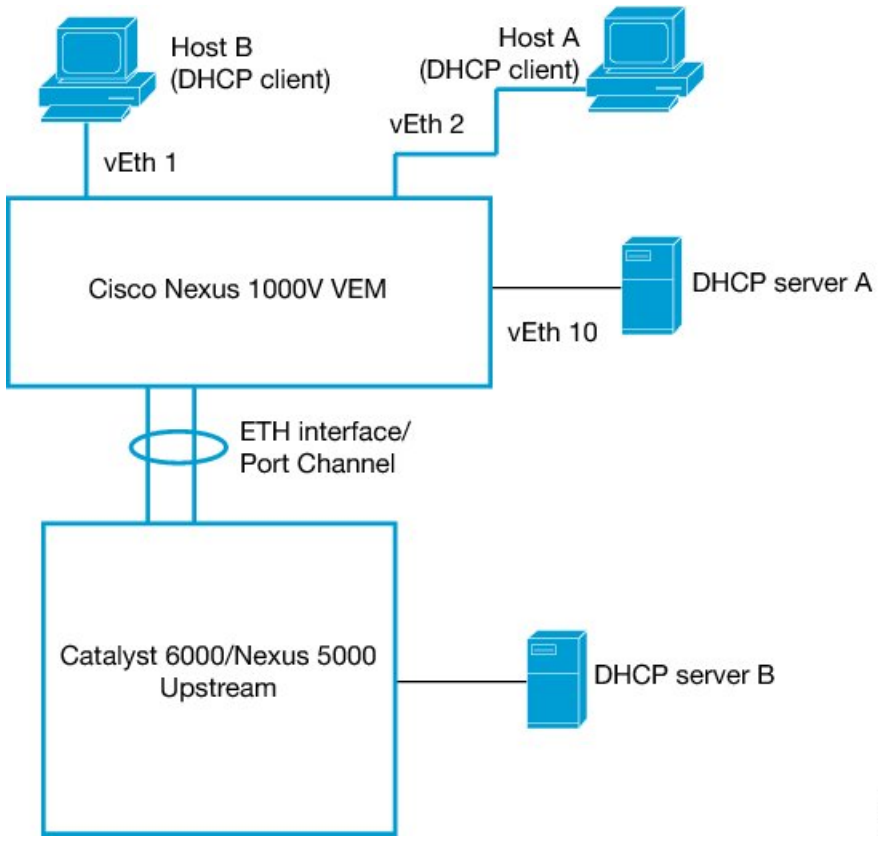
Configuration Example for Trust Configuration and DHCP Server Placement in the Network

DHCP Server Inside and Outside the Cisco Nexus 1000V Network and Clients on the Cisco Nexus 1000V

This example shows that there are two DHCP servers: server A on the Nexus 1000V and Server B on the upstream switch. Clients A and B can get the IP address from DHCP server B without any additional trust configuration because the Ethernet ports/port-channel interface on the Cisco Nexus 1000V are trusted by default.

The following figure shows that to use DHCP server A, you must configure trust on vEthernet 10 to which the server is connected.

Figure 15: DHCP Server Inside and Outside the Cisco Nexus 1000V Network and Clients on the Cisco Nexus 1000V

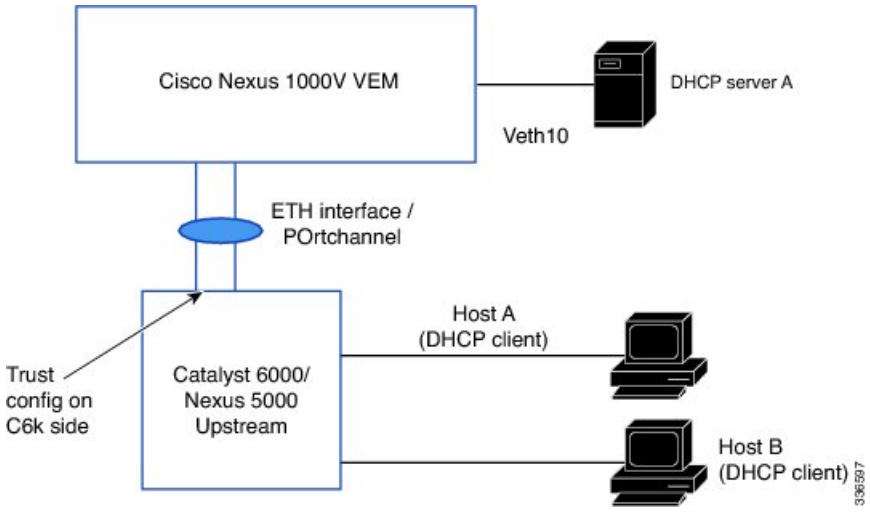


DHCP Server Inside the Cisco Nexus 1000V Network and Clients Outside the Cisco Nexus 1000V

You can configure interfaces on the upstream switch as trusted if the administrator is running the DHCP server on a Virtual Machine (VM) on the Cisco Nexus 1000V and clients are outside the Cisco Nexus 1000V.

In the following figure, server A is on the Cisco Nexus 1000V and clients A and B can get the IP address from server A only when trust is enabled on the ports on the upstream side.

Figure 16: DHCP Server Inside the Cisco Nexus 1000V Network and Clients Outside the Cisco Nexus 1000V



Standards

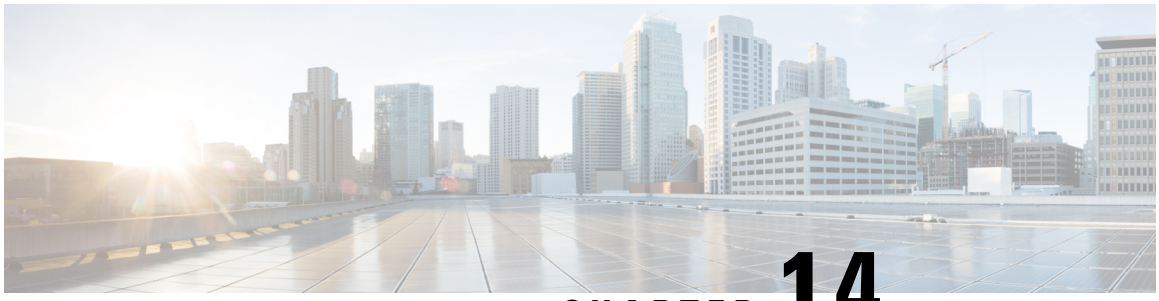
Standards	Title
RFC-2131	Dynamic Host Configuration Protocol (http://tools.ietf.org/html/rfc2131)
RFC-3046	DHCP Relay Agent Information Option (http://tools.ietf.org/html/rfc3046)

Feature History for DHCP Snooping

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
Licensing changes	4.2(1)SV2(1.1)	DHCP snooping is available as an advanced feature. Use the **feature dhcp** command to enable the feature.

Feature Name	Releases	Feature Information
Enabling Source IP Based Filtering	4.2(1)SV2(1.1)	You can enable source IP-based filtering on the Cisco Nexus 1000V switch.
feature dhcp command	4.2(1)SV1(4)	Command added for enabling the DHCP feature globally.
DHCP snooping	4.0(4)SV1(2)	This feature was introduced.



CHAPTER 14

Configuring Dynamic ARP Inspection

This chapter contains the following sections:

Information About Dynamic ARP Inspection

This section provides information about DAI features.

ARP

Dynamic ARP Inspection (DAI) ensures that only valid ARP requests and responses are relayed by intercepting all ARP requests and responses on untrusted ports and verifying 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. When this feature is enabled, invalid ARP packets are dropped.

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.

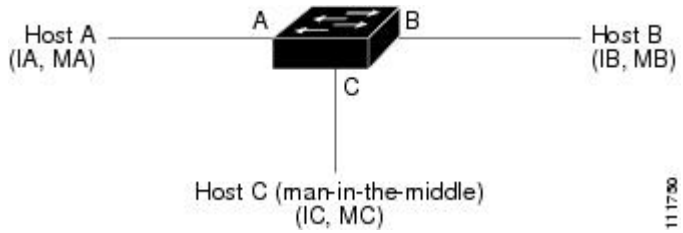
Starting with Release 4.2(1)SV2(1.1), you can filter the traffic based on the source IP address only as opposed to filtering the traffic based on the IP-MAC Address pair. For more information, refer to Enabling Source IP-Based Filtering, on page 194.

ARP Spoofing Attacks

In an ARP spoofing attack, a host allows an unsolicited ARP response to update its cache so that traffic is directed through the attacker until it is discovered and the information in the ARP cache is corrected.

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.

Figure 17: ARP Cache Poisoning



In the figure, 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 add a binding to their ARP caches for a host with the IP address IA and a MAC address MA.

When host B responds, the device and host A update their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can spoof host A and B by broadcasting the following forged ARP responses:

- One for Host B with an sorce IP Address IA and source MAC address MC
- One for Host A with an source IP Address IB and source MAC address MC

Host B then uses MC as the destination MAC address for traffic that was intended for IA, which means that host C intercepts that traffic. Likewise, host A uses MC as destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a man-in-the middle attack.

DAI and ARP Spoofing

DAI is used to validate ARP requests and responses as follows:

- Intercepts all ARP requests and responses on untrusted ports.
- Verifies that a packet has a valid IP-to-MAC address binding before updating the ARP cache or forwarding the packet.
- 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 when it is enabled on the VLANs and on the device. It may also contain static entries that you have created.

If an ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header.

Interface Trust and Network Security

DAI identifies interfaces as trusted or untrusted.

In a typical network, interfaces are configured as follows:

- Untrusted—Interfaces that are connected to hosts.

  Packets are validated by DAI.

- Trusted—Interfaces that are connected to devices.

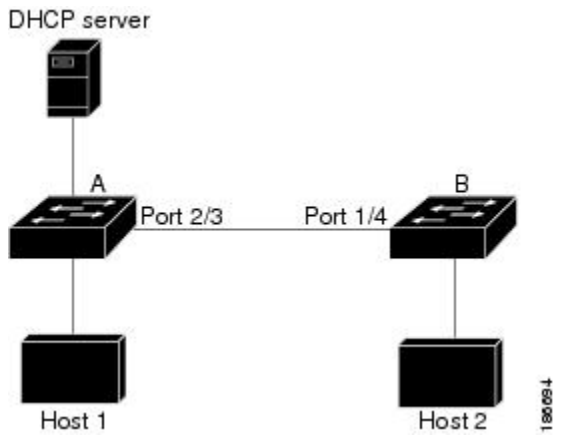
  Packets bypass all DAI validation checks.

With this configuration, all ARP packets that enter the network from a device bypass the security check. No other validation is needed at any other place in the VLAN or in the network.

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In the following figure, 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 18: ARP Packet Validation on a VLAN Enabled for DAI



If you configure interfaces as trusted when they should be untrusted, you might 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.

Prerequisites for DAI

- You must be familiar with the following:
  - ARP
  - DHCP snooping
- The software running on your Cisco Nexus 1000V must support DAI.
- The VEM feature level must be updated to a release that supports DAI.

Guidelines and Limitations for DAI

- DAI is an ingress security feature and does not perform any egress checking.
- DAI is not effective when the host is connected to a device that does not support DAI or that does not have DAI enabled. To prevent attacks that are limited to a single Layer 2 broadcast domain, you should separate a domain with DAI from those 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, you must configure DHCP snooping only. If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, you must configure DHCP snooping on the same VLANs on which you configure DAI.
- DAI is supported on vEthernet interfaces and private VLAN ports
- Virtual Service Domain (VSD) service VM ports are trusted ports by default. Even if you configure VSD ports as untrusted, they still appear as trusted ports to DAI.

Default Settings for DAI

Parameters	Default
VLAN	VLANs are not configured for DAI.
Trust state of vEthernet interfaces not in a VSD	Untrusted.

Parameters	Default
Trust state of vEthernet interfaces in a VSD	Trusted.
Trust state of Ethernet port channels	Trusted.
Incoming ARP packet rate limit for untrusted interfaces	15 packets per second (pps).
Incoming ARP packet rate limit for trusted	15 packets per second (pps).
Rate limit burst interval	5 seconds.
Detecting and recovering DAI error-disabled interfaces	Error-disabled detection and recovery is not configured.
Validation checks (source MAC/ Destination MAC /IP)	No checks are performed.
VLAN statistics	ARP request and response statistics.

Configuring DAI Functionality

Configuring a VLAN for DAI

By default, VLANs are not configured for DAI.

Before you begin

- Log in to the CLI in EXEC mode.
- Enable DHCP snooping.
- Create the VLANs that you want to configure for DAI.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip arp inspection vlan** *list*	Configures the specified VLAN or list of VLANs for DAI.
Step 3	(Optional) switch(config)# **show ip arp inspection vlan** *list*	Displays the DAI status for the specified list of VLANs.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a VLAN for DAI:

```
switch# configure terminal
switch(config)# ip arp inspection vlan 100
switch(config)# show ip arp inspection vlan 100
Source Mac Validation : Disabled
Destination Mac Validation : Disabled
IP Address Validation : Disabled

Filter Mode (for static bindings): IP-MAC

Vlan : 100
-----------
Configuration : Enabled
Operation State : Active
DHCP logging options : Deny
switch(config)# copy running-config startup-config
```

Configuring a Trusted vEthernet Interface

By default, vEthernet interfaces are untrusted, unless they are part of a VSD.

If an interface is untrusted, all ARP requests and responses are verified for a valid IP-MAC address binding before the local cache is updated and the packet forwarded. If a packet has an invalid IP-MAC address binding, it is dropped.

ARP packets that are received on a trusted interface are forwarded but not checked.

You can configure a trusted interface on either of the following:

- The interface itself
- The existing port profile that the interface is assigned to

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified vEthernet interface.
Step 3	switch(config)# **port-profile** *profilename*	Places you in port profile configuration mode for the specified port profile.
Step 4	switch(config-if)# [**no**]**ip arp inspection trust**	The **no** option configures the port as untrusted for ARP inspection.

	Command or Action	Purpose
Step 5	switch(config-port-profile)# **ip arp inspection trust**	Configures the interfaces assigned to the port profile as trusted ARP interfaces.
Step 6	(Optional) switch(config-if)# **show ip arp inspection interface vethernet** *interface-number*	Displays the trusted state and the ARP packet rate for the specified interface.
Step 7	(Optional) switch(config-if)# **show port-profile name** *profilename*	Displays the port profile configuration including the ARP trusted state.
Step 8	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a trusted vEthernet interface:

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# ip arp inspection trust
switch(config-if)# show ip arp inspection interfaces vethernet 3

Interface Trust State Pkt Limit Burst Interval
------------- ----------- --------- --------------
Vethernet3 Trusted 0 0
switch(config-if)# copy running-config startup-config

switch(config-if)# port-profile vm-data
switch(config-port-prof)# ip arp inspection trust
switch(config-port-prof)# show port-profile name vm-data

port-profile vm-data
type: Vethernet
description:
status: enabled
max-ports: 32
min-ports: 1
inherit:
config attributes:
switchport mode access
switchport access vlan 100
ip arp inspection trust
no shutdown
evaluated config attributes:
switchport mode access
switchport access vlan 100
ip arp inspection trust
no shutdown
assigned interfaces:
port-group: vm-data
system vlans: none
capability l3control: no
capability iscsi-multipath: no
capability vxlan: no
capability l3-vservice: no
port-profile role: none
port-binding: static
```

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

Resetting a vEthernet Interface to Untrusted

By default, vEthernet interfaces are untrusted, unless they are part of a VSD. You can remove a trusted designation from a vEthernet interface and return it to the default untrusted designation.

If an interface is untrusted, all ARP requests and responses are verified for a valid IP-MAC address binding before the local cache is updated and the packet forwarded. If a packet has an invalid IP-MAC address binding, it is dropped.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in the interface configuration mode for the specified vEthernet interface.
Step 3	switch(config-if)# **default ip arp inspection trust**	Removes the trusted designation from the interface and returns it to the default untrusted state.
Step 4	(Optional) switch(config-if)# **show ip arp inspection interface vethernet** *interface-number*	Displays the trusted state and the ARP packet rate for the specified interface.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to reset a vEthernet interface to a untrusted state:

```
switch(config-if)# default ip arp inspection trust
switch(config-if)# show ip arp inspection interface vethernet 3
Interface     Trust State Pkt Limit Burst Interval
------------- ----------- --------- ----------------
Vethernet3     Untrusted   15          5
switch(config-if)# copy running-config startup-config
```

Configuring DAI Rate Limits

You can set the rate limit of ARP requests and responses.

Because of their aggregation, trunk ports should be configured with higher rate limits.

Once the rate of incoming packets exceeds the configured rate, the interface is automatically put into an errdisable state.

The default DAI rate limits are as follows:

- Untrusted interfaces—15 packets per second
- Trusted interfaces—Unlimited
- Burst interval—5 seconds

You can configure the rate limits for an interface on either of the following:

- The interface itself
- The existing port profile that the interface is assigned to
- If configuring the port profile, it has already been created and you know its name.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified vEthernet interface.
Step 3	switch(config)# **port-profile** *profilename*	Places you in port profile configuration mode for the specified port profile.

Step 4	switch(config-if)# **ip arp inspection limit** {**rate** *pps* [**burst interval** l *bint*] | **none**}	Configures the specified ARP inspection limit on the interface or the port profile as follows.
		The keywords are as follows:
		• **rate**—Specifies that allowable values are between 1 and 2048 packets per second (pps).
		  • The untrusted interface default is15 packets per second.
		  • The trusted interface default is15 packets per second.
		• **burst interval**—Specifies that allowable values are between 1 and 15 seconds (the default is 5 seconds).
		• **none**—Specifies an unlimited number of packets per second.

	Command or Action	Purpose
Step 5	(Optional) switch(config-if)# **show ip arp inspection interface vethernet interface-number**	Displays the trusted state and the ARP packet rate for the specified interface.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create DAI rate limits:

```
switch# configure terminal
switch(config)#interface vethernet 3
switch(config-if)#ip arp inspection limit rate 30
switch# show ip arp inspection interfaces vethernet 3

Interface Trust State Pkt Limit Burst Interval
------------- ----------- --------- -----------------
Vethernet9 Untrusted    30          5
switch#copy running-config startup-config
```

Resetting DAI Rate Limits to Default Values

You can set the rate limit of ARP requests and responses.

Because of their aggregation, trunk ports should be configured with higher rate limits.

Once the rate of incoming packets exceeds the configured rate, the interface is automatically put into an errdisable state.

The default DAI rate limits are as follows:

- Untrusted interfaces—15 packets per second
- Trusted interfaces—Unlimited
- Burst interval—5 seconds

You can configure the rate limits for an interface on either of the following:

- The interface itself
- The existing port profile that the interface is assigned to

  If configuring the port profile, it has already been created and you know its name.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified vEthernet interface.

Step 3	switch(config-if)# **default ip arp inspection limit** {**rate** *pps* [**burst interval**l *bint*] | **none**}	Removes the configured DAI rate limits from the interface and returns them to the default values.
		The keywords are as follows:
		• **rate**—Specifies that the untrusted interface default is 15 packets per second.
		  • The untrusted interface default is 15 packets per second.
		  • The trusted interface default is 15 packets per second.
		• **burst interval**—Specifies the range is from 1 to 15 seconds. The default is 5 seconds.
		• **none**—Specifies an unlimited number of packets per second.

Step 4	(Optional) switch(config)# **show ip arp inspection interface vethernet** *interface-number*	Displays the default ARP packet rate for the specified interface.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to reset DAI rate limits to their default values:

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# default ip arp inspection limit rate

switch# show ip arp inspection interface vethernet 3
<------no output expected for this, since interface moved to default---->

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

Detecting and Recovering Error-Disabled Interfaces

By default, interfaces are not configured for DAI error-disabled recovery.

To manually recover an interface from the error-disabled state, use the following command sequence.

1. **shutdown**
2. **no shutdown**

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **errdisable detect cause arp-inspection**	Configures the detection of interfaces that have been error-disabled by ARP inspection. The **no** option disables the detection.
Step 3	switch(config)# [**no**] **errdisable recovery cause arp-inspection**	Configures the auto-recovery of interfaces that have been error-disabled by ARP inspection.
Step 4	switch(config)# **errdisable recovery interval** *timer-interval*	Configures the recovery interval for interfaces that have been error-disabled by ARP inspection. The *timer-interval* is from 30 to 65535 seconds.
Step 5	(Optional) switch(config)# **show errdisable detect**	Displays the errdisable configuration.
Step 6	(Optional) switch(config)# **show errdisable recovery**	Displays the errdisable configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to detect and recover error-disabled interfaces:

```
switch# configure terminal
switch(config)# errdisable detect cause arp-inspection
switch(config)# errdisable recovery cause arp-inspection
switch(config)# errdisable recovery interval 30
switch(config)# show errdisable detect
ErrDisable Reason Timer Status
----------------- ------------
link-flap enabled
bpduguard enabled
dhcp-rate-limit enabled
```

```
arp-inspection enabled
ip-addr-conflict enabled
switch(config)# show errdisable recovery
ErrDisable Reason Timer Status
----------------- ------------
link-flap disabled
bpduguard disabled
dhcp-rate-limit enabled
arp-inspection enabled
security-violation disabled
psecure-violation enabled
failed-port-state enabled
ip-addr-conflict disabled

Timer interval: 30
switch(config)# copy running-config startup-config
```

Validating ARP Packets

You can enable validation of the following, which are disabled by default:

- Destination MAC address

  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 address

  Checks the ARP body for invalid and unexpected IP addresses, including 0.0.0.0, 255.255.255.255, and any IP multicast address. Sender IP addresses are checked in both ARP requests and responses. Target IP addresses are checked only in ARP responses.

- Source MAC address

  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.

Note Whenever you configure a validation, any previous validation configuration is overwritten.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}	Enables the specified validation and overwrites any existing validation that was previously saved: • Source MAC • Destination MAC • IP You can specify all three of these validations but you must specify at least one. Use the **no** option to disable a validation.
Step 3	(Optional) switch(config)# **show ip arp inspection**	Displays the DAI configuration.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to validate ARP packets:

```
switch# configure terminal
switch(config)# ip arp inspection
switch(config)# show ip arp inspection
Source Mac Validation      : Enabled
Destination Mac Validation      : Enabled
IP Address Validation      : Enabled

Filter Mode(for static bindings): IP-MAC
switch(config)# copy running-config startup-config
```

Enabling Source IP-Based Filtering

When you assign static IP addresses to virtual machines (VMs) in the deployment and the VMs power on and off frequently, the MAC addresses of the VMs change. This situation affects the Dynamic ARP Inspection (DAI) and the IP Source Guard (IPSG) functionality on the Cisco Nexus 1000V. The Cisco Nexus 1000V does not have the same IP-MAC address binding. Therefore, the traffic from these VMs is dropped.

Starting with Release 4.2(1)SV2(1.1), you can filter the traffic based on the source IP address only. The Cisco Nexus 1000V ignores the MAC address and validates only the source IP address of the traffic from the VMs. This new functionality is applicable to static bindings only.

To enable source IP based filtering on the Cisco Nexus 1000V switch, set the filter mode to ip filtering. The default filtering mode is the ip-mac filtering mode.

Before you begin

- Log in to the CLI in EXEC mode.

- Enable DHCP feature on the Cisco Nexus 1000V switch.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DHCP globally. DHCP snooping is available as an advanced feature that requires a license.
Step 3	switch(config)# **ip source binding filter-mode ip | ip-mac**	Configures the filter mode.
Step 4	(Optional) switch(config)# **show ip source binding filter-mode**	Displays the filter mode on the switch.
Step 5	(Optional) switch(config)# **show ip arp inspection**	Displays the filter mode as part of the output.
Step 6	(Optional) switch(config)# **show ip arp inspection vlan***vlan-id*	Displays the filter mode as part of the output.
Step 7	(Optional) switch(config)# **show ip verify source**	Displays the filter mode as part of the output.
Step 8	(Optional) switch(config)# **show ip verify source interface vethernet** *interface-number*	Displays the filter mode as part of the output.

Example

This example shows how to filter the traffic based on the **IP** filter mode:

```
switch# configure terminal
switch(config)# feature dhcp
switch# show ip source binding filter-mode
DHCP Snoop Filter Mode(for static bindings) = IP-MAC
switch# configure terminal
switch(config)# ip source binding filter-mode ip
switch# show ip source binding filter-mode
DHCP Snoop Filter Mode(for static bindings) = IP
switch# show ip arp inspection

Source Mac Validation      : Disabled
Destination Mac Validation      : Disabled
IP Address Validation      : Disabled

Filter Mode(for static bindings): IP

Vlan : 1
-----------
Configuration          : Enabled
Operation State        : Active
DHCP logging options   : Deny

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

switch# show ip verify source
Filter Mode(for static bindings): IP
IP source guard is enabled on the following interfaces:
-------------------------------------------------------
        Vethernet1
        Vethernet2
        Vethernet3
        Vethernet4
        Vethernet5
        Vethernet6
        Vethernet7
        Vethernet8
        Vethernet9
        Vethernet10


IP source guard operational entries:
------------------------------------
Interface         Filter-mode             IP-address     Mac-address       Vlan
------------      -----------             ----------     --------------    ----
Vethernet1        active                1.182.56.137   00:50:56:82:56:3e  1
Vethernet2        active                1.182.56.138   00:50:56:82:56:3f  1
Vethernet3        active                1.182.56.139   00:50:56:82:56:40  1
Vethernet4        active                1.182.56.140   00:50:56:82:56:41  1
Vethernet5        active                1.182.56.141   00:50:56:82:56:42  1
Vethernet6        active                1.182.56.142   00:50:56:82:56:43  1
Vethernet7        active                1.182.56.143   00:50:56:82:56:44  1
Vethernet8        active                1.182.56.144   00:50:56:82:56:45  1
Vethernet9        active                1.182.56.145   00:50:56:82:56:46  1
Vethernet10       active                1.182.56.146   00:50:56:82:56:47  1
switch#

switch# show ip verify source interface vethernet 1
Filter Mode(for static bindings): IP
IP source guard is  enabled on this interface.

Interface         Filter-mode             IP-address     Mac-address       Vlan
------------      -----------             ----------     --------------    ----
Vethernet1        active                1.182.56.137   00:50:56:82:56:3e    1
```

Verifying the DAI Configuration

Use the following commands to verify the configuration:

Command	Purpose
show running-config dhcp	Displays the DAI configuration.
show ip arp inspection	Displays the status of DAI.

Command	Purpose
show ip arp inspection interface vethernet *interface-number*	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.

Monitoring DAI

Use the following commands to monitor DAI:

Command	Purpose
show ip arp inspection statistics	Displays DAI statistics.
show ip arp inspection statistics vlan *vlan-ID*	Displays DAI statistics for a specified VLAN.
clear ip arp inspection statistics	Clears DAI statistics.

This example shows how to display IP ARP statistics:

```
switch# show ip arp inspection statistics

Vlan : 13
-----------
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

Vlan : 1054
-----------
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

Vlan : 1058
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
```

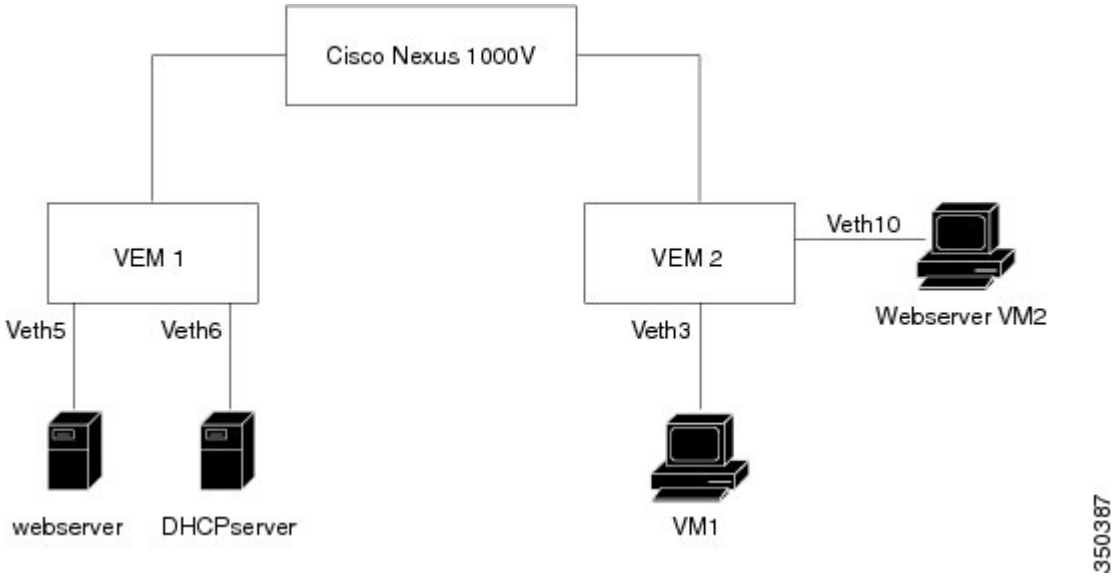
```
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0


switch# show ip arp inspection statistics vlan 13

Vlan : 13
-----------
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
switch#
```

Configuration Examples for DAI

These examples show how to configure DAI in a network with two VEMs:

- One VEM is hosting an authentic web server and a DHCP server.
- The other VEM is hosting a client Virtual Machine (VM 1) and a Virtual Machine (VM 2) with a rogue web server. VM 1 is connected to vEthernet interface 3, which is untrusted by default, and belongs to VLAN 1. VM 2 is connected to vEthernet 10 and VLAN 1.

Figure 19: Configuring DAI in a Network



Without DAI enabled, VM 2 can spoof the ARP cache in VM 1 by sending a packet even though an ARP request was not generated. In this case, the packet directs VM 1 to send its traffic to the VM 2 web server instead of the authentic web server.

If DAI is enabled when VM2 attempts to spoof the ARP cache in VM1, the unsolicited ARP packet sent by VM 2 is dropped because DAI detects the invalid IP-to-MAC address binding. The attempt to spoof the ARP cache fails, and VM 1 connects to the authentic web server.

Note DAI depends on the DHCP snooping 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.

Enabling DAI on VLAN 1 and Verifying the Configuration

This example shows how to enable DAI on VLAN 1 and add a static binding for the web server on interface veth5:

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

switch(config)# ip arp inspection vlan 1

switch# show ip arp inspection vlan 1

Source Mac Validation      : Enabled
Destination Mac Validation      : Enabled
IP Address Validation      : Enabled

Filter Mode (for static bindings): IP-MAC

Vlan : 1
-----------
Configuration        : Enabled
Operation State      : Active
DHCP logging options : Deny

switch(config)# ip arp inspection validate dst-mac src-mac ip

Note: Validate helps in inspecting the dst-mac,src-mac and ip of ARP packet and Ethernet
Header, while sending the ARP packet.

switch(config)# ip source binding 192.168.2.22 00:50:56:1e:2c:1c vlan 1 interface vethernet
 5
switch# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type        VLAN  Interface
-----------------  ---------------  --------  ----------  ----  -------------
00:50:56:1e:2c:1c  22.22.22.23      infinite   static      1     Vethernet5

switch(config)# int vethernet 6
switch(config-if)# ip arp inspection trust

switch# show ip arp inspection interfaces vethernet 6
Interface Trust State Pkt Limit Burst Interval
------------- ----------- --------- -----------------
Vethernet6 Trusted 0 0
```

```
switch(config)# interface vethernet 3
switch(config-if)# ip arp inspection limit rate 20
switch# show ip arp inspection interfaces vethernet 3

 Interface        Trust State        Pkt Limit      Burst Interval
 -------------    -----------        ---------      ----------------
 Vethernet3          Untrusted           20               5

switch(config)# errdisable detect cause arp-inspection

switch# show ip dhcp snooping binding
MacAddress          IpAddress        LeaseSec  Type         VLAN  Interface
------------------  ---------------  --------  -----------  ----  -------------
00:50:56:1e:2c:1c  192.168.2.22     infinite   static       1     Vethernet5
00:50:56:82:56:43  192.168.2.2      infinite   static       1     Vethernet6
00:50:56:82:56:3e  192.168.2.11     9000       dhcp-snoop   1     Vethernet1
00:50:56:82:56:3f  192.168.2.12     9000       dhcp-snoop   1     Vethernet3
00:50:56:82:56:40  192.168.2.13     9000       dhcp-snoop   1     Vethernet10


If the Rouge-server sends an ARP packet with an IP of 192.168.2.22 (IP of the webserver)
and a MAC address of 00:50:56:82:56:40, ARP packet will be
dropped. An error message will be logged as shown below:

2013 Mar  6 03:54:04 switch %DHCP_SNOOP-SLOT130-3-DHCPDENIEDARP: ARP frame denied due to
DHCP snooping binding on interface Veth10 vlan 1 sender
mac 00:50:56:82:56:40 sender ip 192.168.2.22 target mac 00:50:56:82:56:3f target ip
192.168.2.12.

If Veth3 send ARP packets greater than the configured limit, Veth3 will be placed into error
 disabled state with the following message.
2013 Mar  6 05:26:22 switch %DHCP_SNOOP-4-ERROR_DISABLED: Interface Vethernet3 has moved
to error disabled state due to excessive rate 20 of
ingress ARP packets
```

Example of Displaying the Statistics for DAI

This example shows how to display the statistics for DAI:

```
switch# show ip arp inspection statistics vlan 1
switch#

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
switch#
```

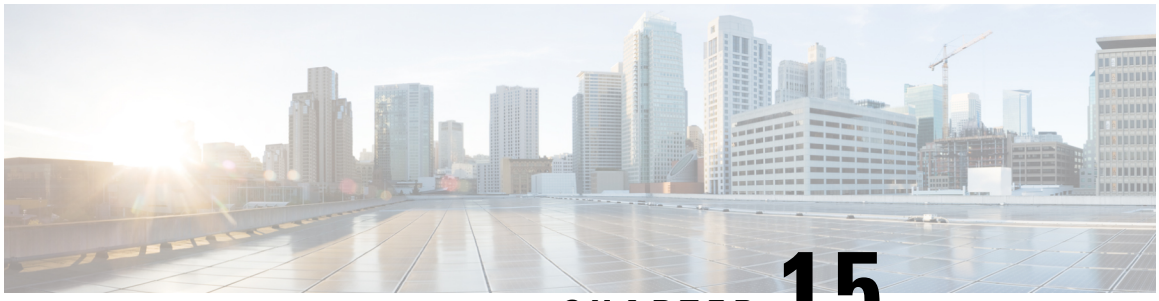
Standards

Standards	Title
RFC-826	An Ethernet Address Resolution Protocol http://tools.ietf.org/html/rfc826

Feature History for DAI

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
Licensing changes	4.2(1)SV2(1.1)	DAI is available as an advanced feature. Use the **feature dhcp** command to enable the feature.
Enabling source IP-based filtering	4.2(1)SV2(1.1)	You can enable source IP-based filtering on the Cisco Nexus 1000V switch.
DAI	4.0(4)SV1(2)	This feature was introduced.



CHAPTER 15

Configuring IP Source Guard

This chapter contains the following sections:

Information About IP Source Guard

IP Source Guard (IPSG) is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches the IP and MAC address bindings of dynamic or static IP source entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table. This feature enables you to control the egress network traffic at the source point. You can configure IPSG in two modes: IP-only mode and IP-MAC mode. The IP-only mode allows you to filter the traffic based on the IP address. The IP address and MAC address combination is used to filter traffic in the IPSG IP-MAC mode. Starting with Cisco Nexus 1000V switch, Release 5.2(1)SV3(2.1), you can now bind multiple IP addresses to a single MAC address for traffic filtering. The multi-IP per MAC functionality enables you to manage traffic from multiple trusted VLANs in a network.

IPSG multi-IP per MAC feature is required to manage traffic when multiple IP addresses are originating from the same interface. For example, you need IPSG multi-IP per MAC feature to source guard a router configured behind a Nexus 1000V switch on a virtual ethernet (veth) trunk port.

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 a source whose static IP entries are configured in the Cisco Nexus 1000V.

The device permits IP packets if the IP address and MAC address of the packet matches a binding table entry or a static IP source entry in the DHCP binding table.

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      vEthernet3
```

If the device receives an IP packet with an IP address of 10.5.5.2, IP Source Guard forwards the packet only if the MAC address of the packet is 00:02:B3:3F:3B:99.

Starting with Release 4.2(1)SV2(1.1), you can filter the IP traffic based on the source IP address only as opposed to filtering the traffic based on the IP-MAC Address pair. For more information, refer to Enabling Source IP-Based Filtering, on page 194.

Prerequisites for IP Source Guard

- You should be familiar with DHCP snooping before you configure IP Source Guard.
- DHCP snooping is enabled.

Guidelines and Limitations for IP Source Guard

- 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 might experience disruption in the IP traffic until the hosts on the interface receive a new IP address from a DHCP server.
- When the IP Source Guard (IPSG) functionality is enabled on the Cisco Nexus 1000V switch and whenever a duplicate IP address is detected on a port, it is error-disabled.
- 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.
- For seamless IP Source Guard, Virtual Service Domain (VSD) service VM ports are trusted ports by default. If you configure these ports as untrusted, this setting is ignored.
- You can attach a maximum of 30 static IP addresses to a single MAC address with mult-IP-per-MAC feature enabled.
- Multi-IP per MAC feature is supported only for static IPSG entries in the DHCP snooping table.

Default Settings for IP Source Guard

Parameters	Default
IP Source Guard	Disabled on each interface.

Parameters	Default
IP source entries	None. No static or default IP source entries exist by default.

Configuring IP Source Guard Functionality

Enabling or Disabling IP Source Guard on a Layer 2 Interface

By default, IP Source Guard is disabled on all interfaces. You can configure IP Source Guard on either an interface or a port profile.

Before you begin

Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters interface configuration mode, where *interface-number* is the vEthernet interface that you want to configure as trusted or untrusted for DHCP snooping.
Step 3	switch(config)# **port-profile** *profilename*	Places you in port profile configuration mode for the specified port profile.
Step 4	switch(config-if)# [**no**] **ip verify source dhcp-snooping-vlan**	Enables IP Source Guard on the interface. The **no** option disables IP Source Guard on the interface.
Step 5	(Optional) switch(config-if)# **show ip verify source interface vethernet interface number**	Displays the IP Source Guard configuration.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable IP Source Guard on a Layer 2 interface:

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# ip verify source dhcp-snooping-vlan
switch (config-if)# show ip verify source interface vethernet 3

Filter Mode(for static bindings): IP-MAC
IP source guard is  enabled on this interface.
```

```
Interface         Filter-mode             IP-address     Mac-address       Vlan
----------    -----------             ----------     -----------    ----
Vethernet3        active                   1.182.56.137   00:50:56:82:56:3e  1053
```

Configuring Multi-IP per MAC feature

Use this procedure to configure multi-IP per MAC feature on IPSG on an interface.

Before you begin

Before beginning this procedure, you must know or do the following:

- Ensure that IP Source Guard feature is enabled.
- Ensure that DHCP snooping is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enters global configuration mode.
Step 3	switch(config)# **ip source binding allow multi-ip-per-mac**	Enables multi-IP per MAC address functionality.
Step 4	switch(config)# **ip source binding** *ip_address mac_address* **vlan** *vlan_Number***interface vethernet***vethernet_number*	Enables multi-IP per MAC address functionality.
Step 5	switch(config)# **port-profile port_profile_Name**	Enables multi-IP per MAC address functionality.
Step 6	Required: switch(config-port-prof))# **ip verify source dhcp-snooping-vlan**	Copies the running configuration to the startup configuration.
Step 7	Required: switch(config-port-prof))# **end**	Copies the running configuration to the startup configuration.
Step 8	switch(config)# **copy running-config start-config**	(Optional) Displays the running configuration for DHCP snooping, including the IP Source Guard configuration.
Step 9	switch(config)# **show running-config dhcp**	(Optional) Displays the running configuration for DHCP snooping, including the IP Source Guard configuration.

Example

The following example shows how to configure multi-IP per MAC feature on IPSG:

```
switch# configure terminal
switch(config)# feature dhcp
switch(config)# ip source binding 1.1.1.1 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# ip source binding 1.1.1.2 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# ip source binding 1.1.1.3 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# port-profile port_profile_1
switch(config-port-prof)# ip verify source dhcp-snooping-vlan
switch(config-port-prof)# end
switch(config)# copy running-config startup-config
switch(config)#
```

Configuration Example for IP Source Guard

This example shows how to create a static IP source entry and then how to enable IP Source Guard on an interface.

```
switch# configure terminal
switch(config)# ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface vethernet 3
switch(config)# interface Vethernet 3
switch(config)# ip verify source dhcp-snooping-vlan
switch(config-port-prof)# show ip verify source interface vethernet 3
Filter Mode(for static bindings): IP-MAC
IP source guard is  enabled on this interface.

Interface         Filter-mode             IP-address     Mac-address       Vlan
------      -----------             ----------     --------------    ----
Vethernet3        active              10.5.22.17     00:1f:28:bd:00:13  100
```

Configuration Example for Multi-IP per MAC Support

The following example shows how to configure multi-IP per MAC support on IP Source Guard on an interface:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# feature dhcp
switch(config)# ip source binding allow multi-ip-per-mac
switch(config)# ip source binding 1.1.1.1 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# ip source binding 1.1.1.1 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# ip source binding 1.1.1.1 0050.5695.ae38 vlan 2611 interface vethernet 1
switch(config)# port-profile port_profile_1
switch(config-port-prof)# ip verify source dhcp-snooping-vlan
switch(config-port-prof)# end
switch(config)# copy running-config startup-config
switch(config)#
```

Verifying the IP Source Guard Configuration

Use the following commands to display and verify the IPSG configuration:

Command	Purpose
show running-config dhcp	Displays DHCP snooping configuration, including the IP Source Guard configuration.

Command	Purpose
show ip verify source	Displays IP-MAC address bindings.
Show ip source binding filter-mode	Displays IPSG filtering mode configured on the interface.
Show ip dhcp snooping binding static	Displays IPSG static entries in DHCP snooping table.

The following example displays the DHCP snooping configuration including IPSG configuration:

```
Nexus-1000v# show running-config dhcp
!Command: show running-config dhcp
!Time: Tue Jun 21 10:30:16 2016

version 5.2(1)SV3(2.1)
feature dhcp


interface Vethernet1
  ip verify source dhcp-snooping-vlan
ip dhcp snooping
ip dhcp snooping vlan 2611
ip source binding allow multi-ip-per-mac
no ip dhcp relay
ip source binding 1.1.1.1 0050.5695.ae38 vlan 2611 interface Vethernet1
ip source binding 1.1.1.2 0050.5695.ae38 vlan 2611 interface Vethernet1
ip source binding 1.1.1.3 0050.5695.ae38 vlan 2611 interface Vethernet1
```

The following example displays the multi-IP per MAC support configuration on IP Source Guard on an interface:

```
Nexus-1000v# sh ip verify source
Filter Mode(for static bindings): IP-MAC
IP source guard is enabled on the following interfaces:
-------------------------------------------------------
        Vethernet1


IP source guard operational entries:
------------------------------------
Interface         Filter-mode             IP-address     Mac-address       Vlan
------------      -----------             ----------     --------------    ----
Vethernet1        active                  1.1.1.1        00:50:56:95:ae:38  2611
Vethernet1        active                  1.1.1.2        00:50:56:95:ae:38  2611
Vethernet1        active                  1.1.1.3        00:50:56:95:ae:38  2611
```

The following example displays IP Source Guard filtering mode configured on an interface:

```
Nexus-1000v# sh ip source binding filter-mode
DHCP Snoop Filter Mode(for static bindings) = IP-MAC
DHCP Snoop Multi IP Addresses Per MAC(for static bindings)= Allowed
Nexus-1000v#
```

The following example displays IP Source Guard static entries in DHCP snooping table:

```
MacAddress          IpAddress        LeaseSec     Type       VLAN  Interface
------------------  ---------------  -----------  --------   ----  -------------
00:50:56:95:ae:38          1.1.1.1          infinite    static      2611  Vethernet1
00:50:56:95:ae:38          1.1.1.2          infinite    static      2611  Vethernet1
00:50:56:95:ae:38          1.1.1.3          infinite    static      2611  Vethernet1
Nexus-1000v#
```

Monitoring IP Source Guard Bindings

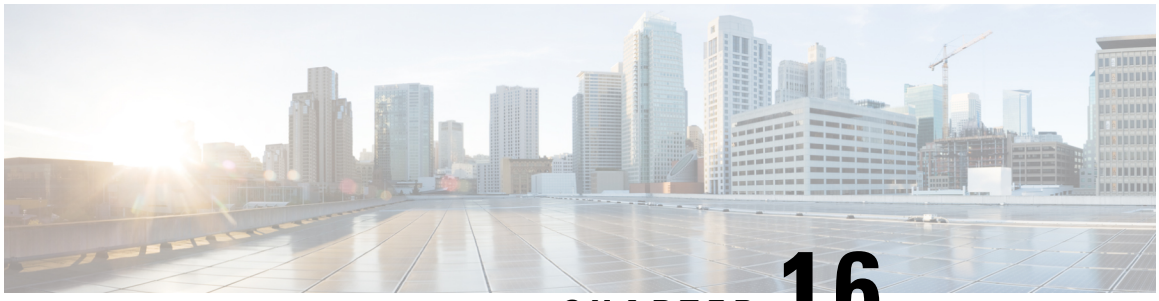
Use the following command to monitor IP Source Guard Bindings.

Command	Purpose
show ip verify source	Displays IP-MAC address bindings

Feature History for IP Source Guard

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
Mulit-IP per MAC Support	5.2(1)SV3(2.1)	Bind multiple IP addresses to a single MAC address for traffic filtering.
Licensing Changes	4.2(1)SV2(1.1)	IP Source Guard is available as an advanced feature. Use the **feature dhcp** command to enable the feature.
Enabling Source IP Based Filtering	4.2(1)SV2(1.1)	You can enable source IP-based filtering on the Cisco Nexus 1000V switch.
IP Source Guard	4.0(4)SV1(2)	This feature was introduced.



CHAPTER 16

Disabling the HTTP Server

This chapter contains the following sections:

Information About the HTTP Server

An HTTP server, which can be turned off from the CLI to address security concerns, is embedded in the Virtual Supervisor Module (VSM).

Guidelines and Limitations for the HTTP Server

- The HTTP server is enabled by default.
- The VMware Update Manager (VUM) does not install Virtual Ethernet Modules (VEMs) if the HTTP server is disabled. During VEM installation, VUM talks directly to the HTTP server to extract required module information from the VSM. To install VEMs, you must do one of the following:
  - Use the VUM by enabling the HTTP server during VEM installation, and then disabling it after the VEMs are installed.
  - Install VEMs manually without using the VUM
- The HTTP server must be enabled in order to get the Cisco Nexus 1000V XML plugin from the VSM.

Default Settings for the HTTP Server

The HTTP server is enabled by default.

Disabling the HTTP Server

By default, the HTTP server is enabled.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature http-server**	Disables the HTTP server.
Step 3	(Optional) switch(config)# **show http-server**	Displays the HTTP server configuration (enabled or disabled).
Step 4	(Optional) switch(config) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to disable the HTTP server:

```
switch# configure terminal
switch(config)# no feature http-server
switch(config)# show http-server
http-server disabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Disabling HTTPS

Before you begin

- Ensure that feature http-server is enabled.
- Ensure that vnm-pa is uninstalled and nsmgr is disabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **http-server no https**	Disables the HTTPS service.
Step 3	(Optional) switch(config)# **show http-server**	Displays the HTTP server configuration.
Step 4	(Optional) switch(config)# **Show feature**	Displays the state (enabled or disabled) of each available feature.

Example

```
switch# configure terminal
switch(config)# http-server no https
httpd: no process killed
switch(config)# show http-server
http-server enabled
 http protocol enabled
 https protocol disabled
switch(config)# show feature
Feature Name                  Instance  State
----------------------------  --------  --------
http-server                   1         enabled
.
.
.
switch(config)#
```

Verifying the HTTP Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show http-server	Displays the HTTP server configuration.
show feature	Displays the features available, such as LACP, and whether they are enabled.

Related Documents for the Disabling the HTTP Server

Related Topic	Document Title
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference*

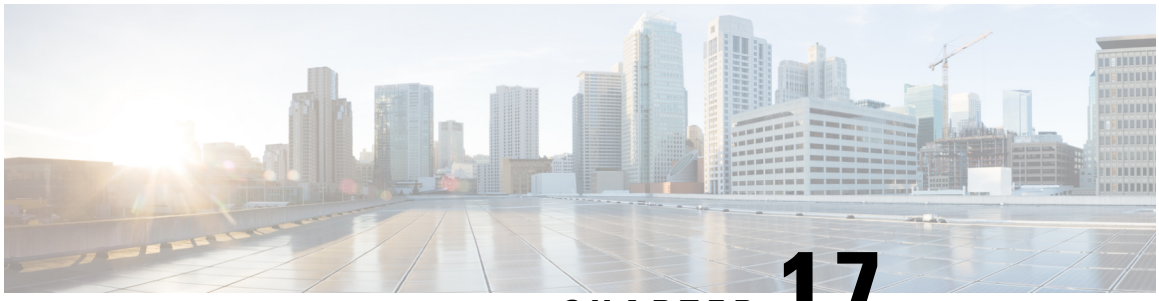
Standards

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 Disabling the HTTP Server

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
Disable HTTP server	4.2(1)SV1(4)	This feature was introduced.



CHAPTER 17

Blocking Unknown Unicast Flooding

This chapter contains the following sections:

Information About UUFB

Unknown unicast packet flooding (UUFB) limits unknown unicast flooding in the forwarding path to prevent the security risk of unwanted traffic reaching the Virtual Machines (VMs). UUFB prevents packets received on both vEthernet and Ethernet interfaces destined to unknown unicast addresses from flooding the VLAN. When UUFB is applied, Virtual Ethernet Modules (VEMs) drop unknown unicast packets received on uplink ports, while unknown unicast packets received on vEthernet interfaces are sent out only on uplink ports.

Guidelines and Limitations for UUFB

- Before configuring UUFB, make sure that the VSM HA pair and all VEMs have been upgraded to the latest release by entering the **show module** command.
- You must explicitly disable UUFB on virtual service domain (VSD) ports. You can disable UUFB in the VSD port profiles.
- You must explicitly disable UUFB on the ports of an application or VM by using MAC addresses other than the one given by VMware.
- Unknown unicast packets are dropped by Cisco UCS fabric interconnects when Cisco UCS is running in end-host-mode.
- On Microsoft Network Load Balancing (MS-NLB) enabled vEthernet interfaces (by entering the **no mac auto-static-learn** command), UUFB does not block MS-NLB related packets. In these scenarios, UUFB can be used to limit flooding of MS-NLB packets to non-MS-NLB ports within a VLAN.

Default Settings for UUFB

Parameters	Default
uufb enable	Disabled
switchport uufb disable	Disabled

Configuring UUFB

Blocking Unknown Unicast Flooding Globally on the Switch

You can globally block unknown unicast packets from flooding the forwarding path for the switch.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enables global configuration mode.
Step 2	switch(config)# [**no**] **uufb enable**	Configures UUFB globally for the VSM.
Step 3	(Optional) switch(config)# **show uufb status**	Displays the UUFB global setting for the VSM.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to block unknown unicast flooding globally:

```
switch# configure terminal
switch(config)# uufb enable
switch(config)# show uufb status
UUFB Status: Enabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Configuring an Interface to Allow Unknown Unicast Flooding

You can allow unknown unicast packets to flood a vEthernet interface if you have blocked flooding globally for the VSM. You can also make sure unknown unicast packets are never blocked on a specific interface, regardless of the global setting.

If you have previously blocked unknown unicast packets globally, you can allow unicast flooding on either a single interface or all interfaces in a port profile.

Before you begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in interface configuration mode for the specified interface.
Step 3	switch(config)# [**no**] **switchport uufb disable**	Disables blocking of unicast packet flooding for the named interface.
Step 4	(Optional) switch(config)# **show running-config vethernet** *interface-number*	Displays the running configuration for the interface for verification.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure an interface to allow unknown unicast flooding:

```
switch# configure terminal
switch(config)# interface vethernet 100
switch(config-if)# switchport uufb disable
switch(config-if)# show running-config interface veth100

!Command: show running-config interface Vethernet100
!Time: Fri Jun 10 12:43:53 2011

version 4.2(1)SV1(4a)

interface Vethernet100
  description accessvlan
  switchport access vlan 30
  switchport uufb disable
switch(config-if)# copy running-config startup-config
[########################################] 100%
```

Configuring a Port Profile to Allow Unknown Unicast Flooding

You can allow unknown unicast packets to flood the interfaces in an existing vEthernet port profile if you have disabled unicast flooding globally for the VSM. You can also make sure unknown unicast packets are never blocked on a specific port profile, regardless of the global setting.

If you have previously blocked unknown unicast packets globally, you can then allow unicast flooding on either a single interface or all interfaces in a port profile.

Before you begin

- Log in to the CLI in EXEC mode.
- Configure the vEthernet port profile for which you want to allow flooding.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *profile-name*	Places you in configuration mode for the named port profile.
Step 3	switch(config-port-prof)# [**no**] **switchport uufb disable**	Disables blocking of unicast packet flooding for all interfaces the named port profile.
Step 4	(Optional) switch(config-port-prof)# **show running-config port-profile** *profile-name*	Displays the configuration for the named port profile for verification.
Step 5	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a port profile to allow unknown unicast flooding:

```
switch# configure terminal
switch(config)# port-profile accessprof
switch(config-port-prof)# switchport uufb disable
switch(config-port-prof)# show running-config port-profile accessprof

!Command: show running-config port-profile accessprof
!Time: Fri Jun 10 12:06:38 2011

version 4.2(1)SV1(4a)
port-profile type vethernet accessprof
  vmware port-group
  switchport mode access
  switchport access vlan 300
  switchport uufb disable
  no shutdown
  description all_access
switch(config-port-prof)# copy running-config startup-config
[########################################] 100%
```

Configuration Example for Blocking Unknown Unicast Packets

This example shows how to block unknown unicast packets from flooding the forwarding path globally for the VSM:

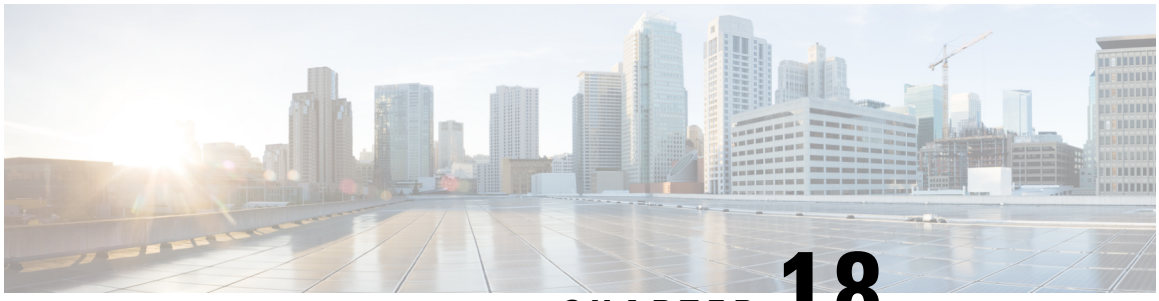
```
n1000v# config terminal
n1000v(config)# uufb enable
n1000v(config)# show uufb status
```

```
UUFB Status: Enabled
n1000v(config)# copy running-config startup-config
[########################################] 100%
```

Feature History for UUFB

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name	Releases	Feature Information
UUFB	4.2(1)SV1(4a)	This feature was introduced.



CHAPTER 18

Configuring Cisco TrustSec

This chapter contains the following sections:

Information About Cisco TrustSec

Cisco TrustSec Architecture

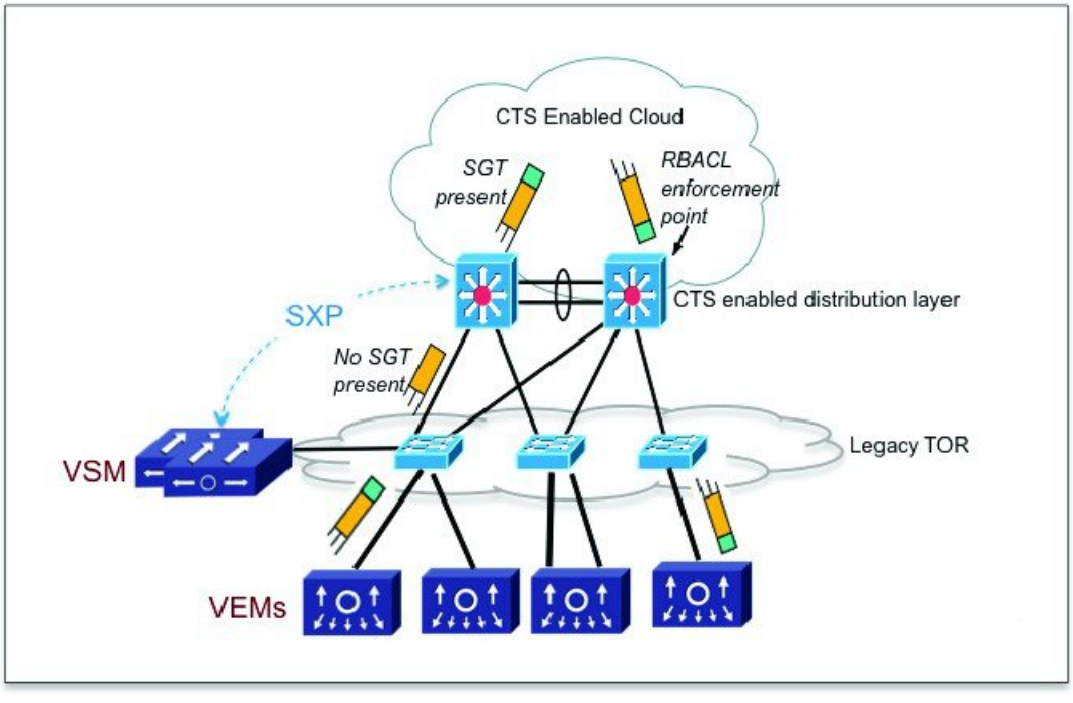
The Cisco TrustSec security architecture enables you to build secure networks by establishing clouds of trusted network devices. Each device in the cloud is authenticated by its neighbors.

Cisco TrustSec uses the device and user identification information that is acquired during authentication to classify or tag packets as they enter the network. These packets are tagged on ingress to the Cisco TrustSec network so that they can be 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 when a packet enters the first Cisco TrustSec-capable device on its path to the destination. Egress refers to when a packet leaves the last Cisco TrustSec-capable device on the path.

This figure shows an example of a Cisco TrustSec cloud.

Figure 20: Cisco TrustSec Network Cloud Example



The Cisco TrustSec architecture consists of the following major components:

- Authentication—Verifies the identity of each device before allowing it to join the Cisco TrustSec network.
- Authorization—Decides the level of access to the Cisco TrustSec network resources that is based on the authenticated identity of the device.
- Access control—Applies access policies on a per-packet basis using the source tags on each packet.
- Secure communication—Provides encryption, integrity, and data-path replay protection for the packets that flow over each link in the Cisco TrustSec network.

Security Group-Based Access Control

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 Cisco NX-OS device, you 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.

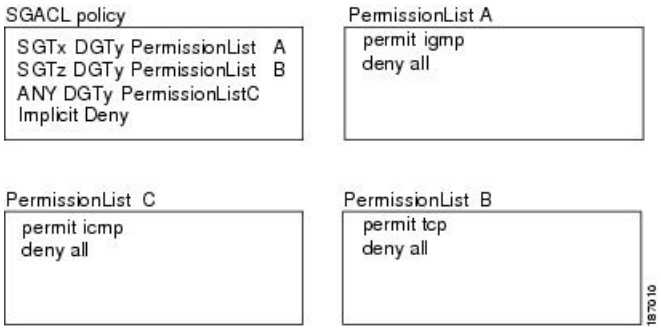
Cisco TrustSec assigns a unique 16-bit tag, called the security group tag (SGT), to a security group. The number of SGTs in the Cisco NX-OS device is limited to the number of authenticated network entities. The SGT is a single label that indicates the privileges of the source within the entire enterprise.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

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

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

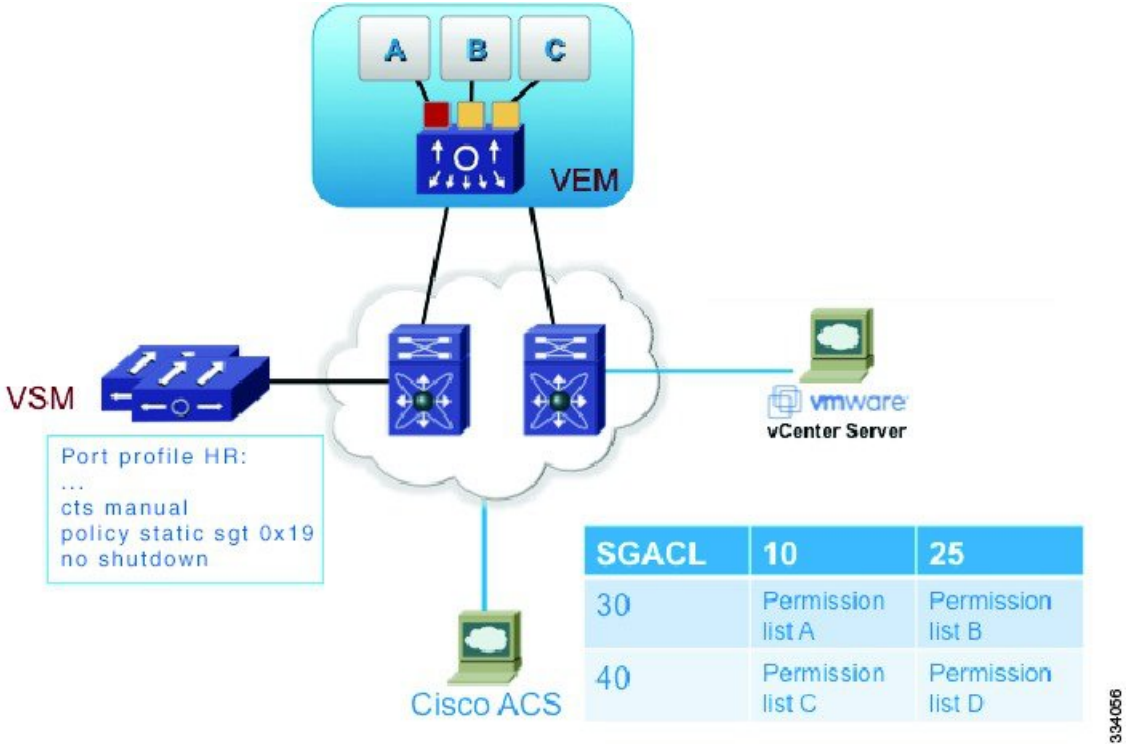
Cisco TrustSec defines access control policies between the security groups. By assigning devices within the network to security groups and applying access control between and within the security groups, Cisco TrustSec achieves access control within the network. The following figure shows an example of an SGACL policy.

Figure 21: SGACL Policy Example



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

Figure 22: SGT and SGACL in Cisco TrustSec Network



The Cisco 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. Cisco TrustSec greatly reduces the size of ACLs and simplifies their maintenance.

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

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

Cisco TrustSec uses the following formula:

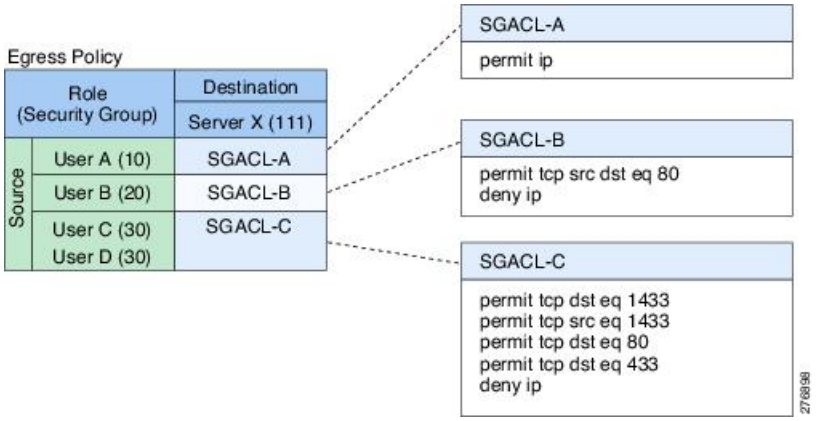
The number of ACEs = The number of permissions specified

SGACL Policies

Using security group access control lists (SGACLs), you can control the operations that users can perform based on the security group assignments of users and destination resources. Policy enforcement within the Cisco TrustSec domain is represented by a permissions matrix, with source security group numbers on one axis and destination security group tags on the other axis. Each cell in the body of the matrix can contain an ordered list of SGACLs which specifies the permissions that should be applied to packets originating from the source security group and destined for the destination security group.

The following figure shows an example of a Cisco TrustSec permissions matrix for a simple domain with three defined user roles and one defined destination resource. Three SGACL policies control access to the destination server based on the role of the user.

Figure 23: SGACL Policy Matrix Example



By assigning users and devices within the network to security groups and applying access control between the security groups, Cisco TrustSec achieves role-based topology-independent access control within the network. Because SGACLs define access control policies based on device identities instead of IP addresses as in traditional ACLs, network devices are free to move throughout the network and change IP addresses. As long as the roles and the permissions remain the same, changes to the network topology do not change the security policy. When a user is added to the switch, you simply assign the user to an appropriate security group and the user immediately receives the permissions of that group.

Using role-based permissions greatly reduces the size of ACLs and simplifies their maintenance. With Cisco TrustSec, the number of access control entries (ACEs) configured is determined by the number of permissions specified, resulting in a much smaller number of ACEs than in a traditional IP network.

Determining the Source Security Group

A network device at the ingress of the Cisco TrustSec cloud needs to determine the SGT of the packet that enters 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 the SGT of the packet so that it can apply the SGACLs.

The network device determines the SGT for a packet in the following order:

1. CTS tag
2. CLI
3. SXP
4. Interface local

Determining the Destination Security Group

The egress network device in a Cisco TrustSec cloud determines the destination group (DGT) for applying the RBACL. This DGT is obtained from the tag that is configured on the egress interface by the interface's port profile.

The network device determines the SGT for a packet in the following order:

1. CLI
2. SXP
3. Interface local

SGACL Enforcement

You configure SGACL enforcement on a port profile. If SGACL enforcement is enabled on the egress interface, the RBACL configured for the (SGT, DGT) pair is applied to the packet. If the packet is dropped, statistics are updated on the ACE. If the SGT is unknown (0), the (*,DGT) policy is applied.

SGT Propagate

You configure SGT Propagate on a port profile. When a packet egresses an interface that is configured with SGT Propagate, the outgoing packet is tagged with a CTS header that carries the SGT for the packet.

Cisco TrustSec With SXPv3

The Security Group Tag (SGT) Exchange Protocol (SXP) is a control protocol that propagates IP address-SGT binding information across network devices. Starting with Cisco Nexus 1000V for VMware vSphere, Release 5.2(1)SV3(2.1), the Cisco TrustSec supports SXP version 3 (SXPv3) to enable transporting IPv4 subnet to the SGT bindings.

By using the subnet-to-SGT bindings, you can minimize the forward information base (FIB) entries needed for storing the mapping, thereby increasing the scale of TrustSec deployments. In many scenarios, you can use subnet-SGT bindings instead of the L3 interface-SGT.

Note SXPv3 does not support IPv6.

SXPv3 Subnet Expansion

The SXPv3 protocol allows you to configure the expansion limit for a subnet binding. SXP expands a subnet binding to host address bindings when a connection is set up with a peer with a version earlier than Version 3. SXP binding expansion applies only to IPv4 subnet binding.

The characteristics of subnet expansion are as follows:

- When expanding the bindings for overlapping IP addresses with different SGT values, the mapping is obtained from the IP address with the longest prefix length.
- If the subnet expansion reaches the configured limit, a system log is generated for the subnet that cannot be expanded.
- Binding expansion does not expand broadcast IP addresses in a subnet. Also, note that SXP does not summarize host IP addresses to subnet bindings. In the SXP propagation path, if there is a node that does not understand subnet binding, the bindings are expanded and propagated through the rest of the propagation path as the host IP binding, even though there is a node that understands subnet binding.
- The default expansion limit is zero (0) and the maximum allowed expansion limit is 4096. You can set the expansion limit as 0 when you do not have any devices in the network that support a lower version of SXP.

You can use the **cts sxp mapping network-map [***num_bindings***]** command to expand the network limit. The *num_bindings* parameter accept a value from 0 to 4096. The value zero (0) indicates that no expansion is allowed and 4096 is the maximum expansion limit allowed. The default value is zero (0).

Consider an example when the expansion limit is set to 67 and the subnet is /24. Cisco NX-OS expands the first 67 IP addresses for the first subnet SGT known to CTS. Since subnet /24 contains more hosts, it will never be fully expanded, and a syslog is generated.

Note When you set the maximum expansion limit as 4096, Cisco NX-OS supports the mapping of every IP in a /16 subnet. However, you must consider the hardware or software impact of setting the expansion limit to the maximum limit.

Cisco TrustSec Subnet-SGT Mapping

The subnet-SGT mapping binds an SGT to all the host addresses of a specified subnet. After this mapping is completed, Cisco TrustSec imposes SGT on the incoming packets with the source IP address that belongs to the specified subnet. This enables you to enforce the CTS policy on the traffic flowing through data center hosts. You can configure IPv4 subnet-SGT bindings under a VRF instance.

A new attribute, *net-mask*, is added to the **cts role-based sgt map** command to define subnet mapping on the VSM.

In IPv4 networks, SXPv3 and later versions can receive and parse subnet network addresses or prefix strings from SXPv3 peers.

For example, the IPv4 subnet 198.1.1.0/29 is expanded as follows (only three bits for host addresses):

- Host addresses 198.1.1.1 to 198.1.1.7 are tagged and propagated to the SXP peer.
- Network and broadcast addresses 198.1.1.0 and 198.1.1.8 are not tagged and not propagated.

Note Use the **cts sxp mapping network-map** global configuration command to limit the number of subnet binding expansions exported to an SXPv1 peer.

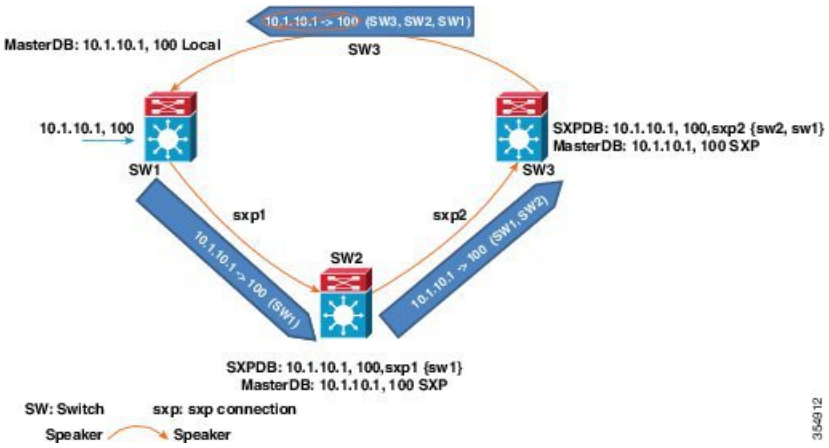
The subnet bindings are static, which means that active hosts are not learned. They can be used locally for SGT imposition and SGACL enforcement. Packets tagged by subnet-SGT mapping can be propagated on Layer 2 or Layer 3 TrustSec links. Additionally, you can use the **cts sxp allow default-route-sgt** command to enable the transport of SGT bindings through the default route, that is, unknown IP address 0.0.0.0.

Overview of Cisco TrustSec with SXPv4

CTS SXP version 4 (SXPv4) enhances the functionality of SXP by adding a loop detection mechanism to prevent stale binding in the network. SXP connections can be enabled such that the binding forwarded by one switch for an SXP connection can be received from another SXP connection, resulting in SXP connection loops. SXP loop topology might, however, result in stale binding in the network. SXPv4' s built-in loop detection and prevention mechanism addresses the stale binding issue whenever there is a loop between SXP nodes.

Loop prevention is achieved by adding SXP propagation path information when propagating (adding/deleting) bindings. Propagation path information keeps track of the network devices (via their node IDs) that the binding travels in an ordered manner. All nodes that participate in the network with looped SXP connections must run SXPv4 to function correctly. Loop detection is a mandatory capability in SXPv4.

Figure 24: SXPv4 Loop Detection



In the figure above there are three network devices: SW1, SW2, and SW3. There are also three SXP connections: SXP1, SXP2 and SXP3, together which create an SXP connection loop. A binding (10.1.10.1, 100) is learned at SW1 through local authentication. The binding is exported by SW1 to SW2 together with the path information (that is, SW1, from where the binding is forwarded).

Upon receiving the binding, SW2 exports it to SW3, again prepending the path information (SW2, SW1). Similarly, SW3 forwards the binding to SW1 with path information SW3, SW2, SW1. When SW1 receives

the binding, the path information is checked. If its own path attribute is in the binding update received, then a propagation loop is detected. This binding is dropped and not stored in the SXP binding database.

If the binding is removed from SW1, (for example, if a user logs off), a binding deletion event is sent. The deletion event goes through the same path as above. When it reaches SW1, no action will be taken as no such binding exists in the SW1 binding database.

Loop detection is done when a binding is received by an SXP but before it is added to the binding database.

The commonly used SXPv4 terms are:

- **SXP Node ID**: SXP Node ID is a 32 bit identifier that is either self-assigned by the switch or router, or can be configured by the user. It is important for the loop detection/prevention functionality.
- **SXP Default Node ID**: If a SXP Node ID is not configured by the user, when SXP is enabled and before establishing a connection, the switch or router has the capability to self-assign the SXP Node ID identifier. For Nexus 1000V, the IP address configured for mgmt0 interface is configured as the default Node ID.
- **SXP Peer sequence**: Sequence of node IDs of the devices through which the IP-SGT binding has traversed in order to reach the listener, with the node ID of the immediate speaker at the head of the list. The peer sequence information is necessary for the accurate loop prevention. The listener discards bindings with its own node ID in the sequence information.
- **SXP Keep-alive mechanism**: In-built keep-alive handshake mechanism between speaker and listener in order to allow for timely detection of connectivity loss, deletion of connection resources and staling of the IP-SGT bindings. SXPV4 capable devices use TCP Keep-alive over V1 and V3 connections.
- **SXP Speaker Hold Time**: The minimum acceptable hold-time that the speaker allows for a connection (directly related to the minimum interval at which speaker will send out keep-alive messages).
- **SXP Listener Hold Time Range**: The hold time range the listener requires for a connection (directly related to the minimum and maximum intervals at which listener expects keep-alive messages from the speaker).
- **SXP Connection Negotiated Hold Time**: The negotiated hold time that the speaker and listener agree upon in the open message hand-shakes prior to connection is established.

Note The listener expects to receive at least one update or keep-alive message within the Listener Hold Time interval on an SXPV4 connection. If the negotiation succeeds, the speaker hold time is lesser than the maximum listener hold time.

- **SXP Capability**: The Nexus 1000V listeners advertises the following capabilities: IPV4 and Subnet-SGT. Additionally, to support default IP-SGT transport in a mixed network, the default IP-SGT capability is exported to allow the speaker to selectively transport default IP-SGT mapping over SXPV4 connections.
- **IP-SGT (Installed) Database**: The installed IP-SGT database that consists of the final IP-SGT bindings amongst all sources (CLI/SXP, Port-sgt etc) that are selected for local installation and transport over SXP.
- **SXP Contributor Database**: This database contains all the host/subnet SGT bindings learnt from every contributor, along with the subsidiary information useful for loop detection and prevention, Peer Sequence:
  - Time-Stamp/ Counter information
  - Active/Contributor status

- SGT and Staling Flags.

- **SXP Contributor Logic**: If there are one or more contributors for the same binding learned at a listener, the SXPV4 listener applies the following logic to determine the active/best SXP contributor:
  - **Shortest Path Rule**: Bindings with the shortest peer-sequence length are preferred.
  - **Most Recently Received Rule**: Bindings learnt most-recently are preferred as a tie-breaker.
- **SXP Version Negotiation**: Refer to the SXP Version Negotiation matrix.

SXP Node ID

An SXP node ID is used to identify the individual devices within the network. The node ID is a four-octet integer that can be configured by the user. If it is not configured by the user, Cisco TrustSec assigns the router ID on the default VRF as the node ID, in the same manner that EIGRP generates its router ID, which is the first IP address on Cisco Nexus 1000V series switches.

The SXP loop detection mechanism drops binding propagation packets based on finding its own node ID in the peer sequence attribute. Changing a node ID in a loop detection-running SXP network could break SXP loop detection functionality and therefore needs to be handled carefully.

The bindings that are associated with the original node ID have to be deleted in all SXP nodes before the new node ID is configured. This can be done by disabling the SXP feature on the network device where you desire to change the node ID. Before you change the node ID, wait until the SXP bindings that are propagated with the particular node ID in the path attribute are deleted.

The node ID configuration is blocked or restricted when SXP is in the enabled state. Router-ID changes in the switch does not affect the SXP node ID, while SXP is enabled. A syslog is generated to indicate that the router ID of the system has changed and this may affect SXP loop detection functionality.

Note Disabling the SXP feature brings down all SXP connections on the device.

Keepalive and Hold-Time Negotiation with SXPv4

SXP uses a TCP-based, keepalive mechanism to determine if a connection is live. SXPv4 adds an optional negotiated keepalive mechanism within the protocol in order to provide more predictable and timely detection of connection loss.

SXP connections are asymmetric with almost all of the protocol messages (except for open/open_resp and error messages) being sent from an SXP speaker to an SXP listener. The SXP listener can keep a potentially large volume of state per connection, which includes all the binding information learned on a connection. Therefore, it is only meaningful to have a keepalive mechanism that allows a listener to detect the loss of connection with a speaker.

The mechanism is based on two timers:

- **Hold timer**: Used by a listener for detection of elapsing time without successive keepalive and/or update messages from a speaker.
- **Keepalive timer**: Used by a speaker to trigger the dispatch of keepalive messages during intervals when no other information is exported via update messages.

The hold-time for the keepalive mechanism may be negotiated during the open/open_resp exchange at connection setup. The following information is important during the negotiation:

- A listener may have desirable range for the hold-time period locally configured or have a default of 90 to 180 seconds. A value of 0xFFFF..0xFFFF indicates that the keepalive mechanism is not used.
- A speaker may have a minimum acceptable hold-time period locally configured or have a default of 120 seconds. This is the shortest period of time a speaker is willing to send keepalive messages for keeping the connection alive. Any shorter hold-time period would require a faster keepalive rate than the rate the speaker is ready to support.
- A value of 0xFFFF implies that the keepalive mechanism is not used.
- The negotiation succeeds when the speaker's minimum acceptable hold-time falls below or within the desirable hold-time range of the listener. If one end turns off the keepalive mechanism, the other end should also turn it off to make the negotiation successful.
- The negotiation fails when the speaker's minimum acceptable hold-time is greater than the upper bound of the listener's hold-time range.
- The selected hold-time period of a successful negotiation is the maximum of the speaker's minimum acceptable hold-time and the lower bound of the listener's hold-time range.
- The speaker calculates the keepalive time to one-third of the selected hold-time by default unless a different keepalive time is locally configured.
- Larger Minimum listener hold-time values are recommended on systems with large number of bindings or connections. Also, these values are recommended if there is a requirement to hold the bindings on the listener during network maintenance events.

Bidirectional SXP Support Overview

The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection.

With the support for bidirectional Security Group Tag (SGT) Exchange Protocol (SXP) configuration, a peer can act as both a speaker and a listener and propagate SXP bindings in both directions using a single connection.

The bidirectional SXP configuration is managed with one pair of IP addresses, thereby reducing operational complexity. On either end, only the listener initiates the SXP connection and the speaker accepts the incoming connection.

Figure 25: Bidirectional SXP Connection

In addition, Bi-directional SXP uses the underlying loop-detection benefits of SXPV4 to avoid replay of updates back and forth across the same connection.

Note The peers at each end of the connection must be configured as a bidirectional connection using the **both** keyword. It is an incorrect configuration to have one end configured as a bidirectional connection using the **both** keyword and the other end configured as a speaker or listener (unidirectional connection). The system will not be able to detect the mismatch in configuration leading to unpredictable SXP connectivity.

Guidelines and Limitations for SXPv4

Cisco TrustSec SXPv4 has the following guidelines and limitations:

- The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection.
- IPV6 bindings are not learned or transported by the Cisco Nexus 1000V series switches over SXPV4 connections. However, the SXPv4 peering with speakers transporting IPv6 bindings are still supported.
- Cisco Nexus 1000V series switches only expands Subnet-SGT bindings over SXPv3 connections.
- After upgrading a switch, the switch advertizes the default SXPv4 version. The appropriate connection versions are re-negotiated with the peers.
- Ensure that there are no overlapping node IDs configured in the network or the node IDs that are configured in the network do not overlap with IP addresses used elsewhere in the network.
- Ensure that there are no overlapping IP addresses to avoid unintentional reuse of default node IDs in the network.
- Prior to modifications to IP addresses in the switch or a router, ensure that the old and the new IP addresses have not been used as default node IDs locally or remotely in the network.
- Ensure that the speaker and listener hold-time values per connection or global or default for each speaker-listener pair are compatible.
- Note that using the hold-time value as 65535 on either speaker or listener disables the in-built keep-alive mechanism and avoids the staling of bindings upon connectivity loss on SXPv4 devices. Administrative connection resets are required to clear these bindings.
- When migrating existing uni-directional connections to bi-directional connections, ensure that the global hold times are compatible and the bindings learnt in both directions are within the supported scale limits. Also, ensure that the global or default hold-time values on speaker and listener are compatible, since you cannot configure hold-time values for these connections on a per-connection basis.

SXP Version Negotiation

The SXP session is established between speaker devices and listener devices. By default, the CTS device advertises the highest supported SXP version. The negotiation is made based on the highest common version supported by the speaker and listener devices. A standalone CTS-supported device can establish an SXP session with different versions, with its peer devices, depending on the SXP versions of the peer devices.

The following table provides information about version negotiation for interoperability in different scenarios.

Table 6: SXP Version Negotiation Cases

Case Number	Speaker	Listener	SXP Session Status
	SXPv1	SXPv1	SXPv1 session is established.
	SXPv1	SXPv2	SXPv1 session is established.
	SXPv1	SXPv3	SXPv1 session is established.
	SXPv1	SXPv4	SXPv1 session is established.
	SXPv1	SXPv4	SXPv1 session is established.
	SXPv2 (Not N1KV)	SXPv1	SXPv1 session is established.
	SXPv2	SXPv2	Not possible because a Cisco Nexus 1000V switch does not support SXPv2.
	SXPv2 (Not N1KV)	SXPv3	When the Cisco Nexus 1000V (SXPv3) Listener receives an OPEN RSP from an SXPv2 speaker: **1.** The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. **2.** The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv3. **3.** On receiving an OPEN with SXPv3 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.
	SXPv2 (Not N1KV)	SXPv4	When the Cisco Nexus 1000V (SXPv4) Listener receives an OPEN RSP from an SXPv2 speaker: **1.** The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. **2.** The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv4. **3.** On receiving an OPEN with SXPv4 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.

Case Number	Speaker	Listener	SXP Session Status
	SXPv3	SXPv1	SXPv1 session is established.
	SXPv3	SXPv2	When the Cisco Nexus 1000V (SXPv3) Listener receives an OPEN RSP from an SXPv2 speaker: 1. The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. 2. The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv3. 3. On receiving an OPEN with SXPv1 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.
	SXPv3	SXPv3	SXPv3 session is established.
	SXPv3	SXPv4	SXPv3 session is established.
	SXPv4	SXPv3	SXPv3 session is established.
	SXPv4	SXPv4	SXPv4 session is established.

Authorization and Policy Acquisition

After authentication ends, 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 the SGT configured on the ingress interface. If enforcement is enabled on this interface, the SGACLs that are associated with the peer SGT are downloaded. If the device does not know if the SGACLs are associated with the peer's SGT, the device might 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 indicate 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.

Licensing Requirements for Cisco TrustSec

The following table shows the licensing requirements for this feature:

Feature	License Requirement
Cisco TrustSec	This feature requires an Advanced Services License. See the *Cisco Nexus 1000V License Configuration Guide* for more information on the licensing requirements for Cisco Nexus 1000V.

Prerequisites for Cisco TrustSec

- You must install the Advanced Services license.
- You must enable the 802.1X feature.
- You must enable the Cisco TrustSec feature.
- You must enable the Cisco TrustSec SXP.

Guidelines and Limitations for Cisco TrustSec

- ISE policies do not take precedence over the policies configured locally on the VSM. If you want ISE policies to take precedence, you must remove the locally-configured policy.
- Cisco TrustSec supports only IPv4 addressing.
- Cisco TrustSec tagging is supported on VXLANs but not on the VXLAN gateway.
- To assign an SGT to a VM, you must manually configure SGT in the port profile.
- A maximum of 6000 IP-SGT mappings can be learned system-wide in the DVS. This total is for entries learned through DHCP snooping and device tracking of individual VMs by ARP as well as IP traffic inspection.
- A maximum of 10 IP-SGT bindings can be learned from a single virtual Ethernet interface.
- The IP-SGT mappings can be communicated to up to 64 SXP peer devices.
- Cisco TrustSec does not support 802.1x or data encryption.
- Cisco TrustSec does not support SXPv2 specifications.
- The number of rules per policy is limited to the number of ACL policies that are supported by Cisco Nexus 1000V.

- CTS propage-sgt configuration does not function as expected.

Default Settings

Table 7: Default Cisco TrustSec Settings

Parameters	Default
Cisco TrustSec	Disabled
SXP	Disabled
SXP default password	None
SXP reconcile period	120 seconds
SXP retry period	60 seconds
Device tracking	Enabled
Interface delete hold timer	60 seconds

Configuring Cisco TrustSec

Enabling the Cisco TrustSec Feature

You must enable the 802.1X feature and the Cisco TrustSec feature on the Cisco Nexus 1000V before you can configure Cisco TrustSec.

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

Before you begin

- Log in to the CLI in EXEC mode.
- Ensure that you have installed the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**feature dot1x** **Example:**	Enables the 802.1X feature.

	Command or Action	Purpose
	`switch(config)# feature dot1x`	
Step 3	switch(config)# **[no] feature cts**	Enables (or disables when you use the **no** form) the Cisco TrustSec feature.
Step 4	(Optional) switch(config)# **show cts**	Displays the Cisco TrustSec configuration.
Step 5	(Optional) switch(config)# **show feature**	Displays the enabled status for features.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the Cisco TrustSec feature:

```
switch# configure terminal
switch(config)# feature cts
switch(config)# show cts
CTS Global Configuration
==============================
CTS support : enabled
CTS device identity : not configured
SGT : 0
CTS caching support : disabled

Number of CTS interfaces in
DOT1X mode : 0
Manual mode : 0
switch(config)#

switch(config)# show feature
Feature Name Instance State
-------------------- -------- --------
cts 1 enabled
dhcp-snooping 1 enabled
http-server 1 enabled
lacp 1 disabled
netflow 1 disabled
network-segmentation 1 disabled
port-profile-roles 1 disabled
private-vlan 1 disabled
segmentation 1 disabled
sshServer 1 enabled
tacacs 1 disabled
telnetServer 1 enabled
vtracker 1 disabled
switch(config)#
```

Configuring Cisco TrustSec Device Credentials

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

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

http://www.cisco.com/c/en/us/support/security/identity-services-engine/tsd-products-support-series-home.html

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**cts device-id** *name* **password** *password*	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive.
Step 3	**exit**	Exits global configuration mode.
Step 4	(Optional) **show cts**	Displays the Cisco TrustSec configuration.
Step 5	(Optional) **show cts environment**	Displays the Cisco TrustSec environment data.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure Cisco TrustSec device credentials:

```
switch# configure terminal
switch(config)# cts device-id MyDevice1 password CiscO321
switch(config)# exit
switch# copy running-config startup-config
```

Enabling Cisco TrustSec SXP

You can enable the Cisco TrustSec SXP on the Cisco Nexus 1000V.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts sxp enable**	Enables (or disables when you use the **no** form) the Cisco TrustSec SXP feature. The default is disabled.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the Cisco TrustSec SXP configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the Cisco TrustSec SXP:

```
switch# configure terminal
switch(config)# cts sxp enable
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
```

This example shows how to expand the network limit for SXPv3 subnet expansion:

```
switch# configure terminal
switch(config)# cts sxp enable
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:0
Unsupported SXP version(s):2
vsm-sxpv3(config)#
vsm-sxpv3(config)# cts sxp mapping network-map 255
vsm-sxpv3(config)# sh cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:255
Unsupported SXP version(s):2
vsm-sxpv3(config)#
```

Configuring Cisco TrustSec Device Tracking

You can configure device tracking to enable VM IP address learning by inspecting the Address Resolution Protocol (ARP) and IP traffic on virtual Ethernet ports.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts device tracking**	Enables device tracking on Cisco TrustSec. **Note** The Cisco Nexus 1000V supports tracking of IP addresses from the ARP/IP traffic inspection on the VEMs and from DHCP snooping. Cisco TrustSec device tracking tracks IP addresses using the ARP/IP traffic inspection on the VEMs. To enable Cisco TrustSec device tracking to track IP addresses from DHCP snooping, you must also enable the DHCP snooping feature. By default, device tracking is enabled.
Step 3	(Optional) switch(config)# **show cts device tracking**	Displays the Cisco TrustSec device tracking configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec device tracking:

```
switch# configure terminal
switch(config)# cts device tracking
enabled
switch(config)#
```

Configuring a Default SXP Password

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

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp default password** [**word** | **7**] *password*	Configures the SXP default password using the following options: • **word**—Specifies an unencrypted default password. • **7**—Specifies an encrypted default password. By default, no SXP password is used.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 4	(Optional) switch(config)# **show running-config cts**	Displays the running configuration for Cisco TrustSec.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the default SXP password:

```
switch# configure terminal
switch(config)# cts sxp default password 7 CiscoPassword
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
```

```
Network Map expansion limit:2000
Unsupported SXP version(s):2
```

Configuring a Default SXP Source IPv4 Address

The Cisco NX-OS software uses the default source IPv4 address in all new TCP connections where a source IPv4 address is not specified. The default source IPv4 address must be set to the IPv4 address of the mgmt0 interface. No other source IPv4 address works when configuring an SXP peer connection.

Note There is no effect on existing TCP connections when you configure the default SXP source IPv4 address.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp default password** *password*	Configures the SXP default password.
Step 3	switch(config)# **cts sxp default source-ip** *mgmt0-interface*	Configures the mgmt0 interface as the SXP default source IPv4 address.
Step 4	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the default SXP source IPv4 address:

```
switch# configure terminal
switch# cts sxp default password xyzexy
switch(config)# cts sxp default source-ip  10.78.1.73
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
```

```
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
switch(config)#
```

Configuring Cisco TrustSec SXP Peer Connections

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

Note The SXP source IPv4 address must be configured with the mgmt0 IPv4 address for all SXP connections.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

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

Step 2	switch(config)# [**no**] **cts sxp connection peer** *peer-ip-address* **source** *source-ip-address*] **password** {[**default**] | [**none** [**required**] *password*} [**mode**{**listener** | **speaker**}] **vrf management**	Configures the SXP address connection.
		• **Source**—Specifies the IPv4 address of the source. The default source is the IPv4 address that you configured using the **cts sxp default source-ip** command.
		• **Password**—Specifies the password that SXP should use for the connection using the following options:
		  • **Default**—Uses the default SXP password that you configured using the **cts sxp default password** command.
		  • **None**—Does not use a password.
		  • **Required**—Uses the password specified in the command.
		• **Mode**—Specifies the role of the remote peer device using the following options:

	Command or Action	Purpose
		• **listener**—The Cisco Nexus 1000V acts as the speaker in the connection and the peer is configured as the listener. • **speaker**—The Cisco Nexus 1000V acts as the listener in the connection and the peer is configured as the speaker. • The **vrf management** keywords specify that the Virtual Routing and Forwarding (VRF) to the peer is the management (mgmt0) interface.
Step 3	(Optional) switch(config)# **show cts sxp connection**	Displays the SXP connections and their status.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec peer connections:

```
switch# configure terminal
switch(config)# cts sxp connection peer 1.2.3.4 password none mode listener vrf management
switch(config)# show cts sxp connection

PEER_IP_ADDR VRF PEER_SXP_MODE SELF_SXP_MODE CONNECTION STATE
10.197.130.184 management listener speaker connected
10.197.130.185 management speaker listener connected
```

Configuring SXPv4

Configuring the Node ID of a Network Device

Before you begin

Enable the Cisco TrustSec feature.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure the node ID of a network device:

switch(config)# **cts sxp node-id** {*sxp-node-id* | **interface** *interface-type* | *ipv4-address*}

Note Use the **no** form of this command to delete a node ID.

Step 3 Exit global configuration modes:

switch(config)# **exit**

Step 4 (Optional) Display the node ID of a network device by using one of the following commands:

switch# **show cts sxp sgt-map**

switch# **show run | include node-id**

switch# **show cts sxp sgt-map detail**

Example: Configuring the Node ID of a Network Device

The following running configuration shows how to configure the node ID of a network device. Replace the placeholders with relevant values for your setup.

```
#Node Id in Hexadecimal format
configure terminal
 cts sxp node-id <0x1-0xffffffff>
 exit

#Node Id in IPv4 address format
configure terminal
 cts sxp node-id <172.16.1.3>
 exit
```

The following example shows how to configure node ID as an interface.

```
switch(config)# cts sxp node-id interface ethernet 1/1
```

Note that the specified interface should have a valid IP configuration. Otherwise, you cannot configure the node ID.

The following example shows how to display the node ID.

```
switch(config)# show cts sxp sgt-map
SXP Node ID(configured):0x00006789

switch(config)# show run | include node-id
cts sxp node-id interface Eth1/1
```

Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

Before you begin

Enable the Cisco TrustSec feature.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:

switch(config)# **cts sxp listener hold-time** *minimum-period maximum-period*

The valid range is from 1-65534 seconds. The default hold-time range for a listener is 90-180 seconds.

Note The maximum-period value must be greater than the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device:

switch(config)# **cts sxp speaker hold-time** *minimum-period*

The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration modes:

switch(config)# **exit**

Step 5 (Optional) Display the hold-time configuration value:

switch# **show run | grep speaker**

switch# **show run | grep listener**

Example: Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a listener device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp listener hold-time <100> <200>
 exit
```

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a speaker device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp speaker hold-time <100>
 exit
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
cts sxp speaker hold-time 456

switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

The peer connection must be configured on both 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.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode [[both | local {listener | speaker} | peer {listener | speaker} | listener | speaker] hold-time** *minimum-period maximum-period*] [**vrf** *vrf-name*]]

Configures the CTS-SXP peer address connection.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the default source address, if configured, or the address of the port.

The **password** keyword specifies the password that CTS-SXP uses for the connection using the following options:

- **default**—Use the default CTS-SXP password you configured using the **cts sxp default password** command.
- **none**—A password is not used.

The **mode** keyword specifies the role of the remote peer device:

- **both** — The specified mode refers that the device is both the speaker and the listener in the bidirectional SXP connection.
- **local**—The specified mode refers to the local device.
- **peer**—The specified mode refers to the peer device.
- **listener**— Specifies that the peer device is the listener.
- **speaker**— Specifies that the peer device is the speaker.

The **hold-time** keyword allows you to specify the length of the hold-time period for the speaker or listener device. The valid range is from 0-65534 seconds. The value 0 is the global or default hold-time. You can disable the keep-alive mechanism by specifying the maximum hold-time value as 65535. If the **hold-time** option is not specified, the global hold-time value is used. However, if the global hold-time configuration is missing, the default hold-time is used.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.

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

Note The maximum-period value must be greater than or equal to the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device:

switch(config)# **cts sxp speaker hold-time** *minimum-period*

The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration mode:

switch(config)# **exit**

Step 5 (Optional) Displays CTS-SXP status and connections:

switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

Example: Disabling Keep-Alive Mechanism at Listener and Speaker Devices

The following running configuration shows how to configure the hold-time for the SXPv4 protocol for each connection. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <10.20.2.2> password default mode local speaker hold-time <500>
 exit
```

The following example shows how to display the hold-time for the SXPv4 protocol for a connection.

```
switch(config)# show run cts | include connection
cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker hold-time 113 314
 vrf default

switch-listener(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x14141409
IP-SGT Mappings as follows:
IPv4,SGT : <1.34.56.45/32 , 119>
Vrf      :1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a
IPv4,SGT : <2.3.11.0/28 , 123>
Vrf      :1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a,0e0e0e01
Total number of IP-SGT Mappings: 2

switch # show cts sxp connection detail

----------------------------------------------
Peer IP           :3.1.1.2
VRF               :default
```

```
PEER MODE         :speaker
Connection State  :connected
Version           :4
Node ID           :0x0e0e0e01
Capability        :UNKNOWN
Conn Hold Time    :120 seconds
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
cts sxp speaker hold-time 456

switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

The following example shows how to disable keep-alive mechanism at listener and speaker devices by configuring maximum values for hold-time.

```
switch# configure terminal
switch(config)# cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker
hold-time 65535 65535 vrf default
switch(config)# exit

switch# configure terminal
switch(config)# cts sxp connection peer 4.5.6.7 source 6.7.8.9 password none mode listener
 hold-time 65535 vrf default
switch(config)# exit
```

Configuring Bidirectional SXP Support

Before you begin

Enable the Cisco TrustSec feature.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure the Cisco TrustSec SXP peer address connection for a bidirectional SXP configuration:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode both** [**vrf** *vrf-name*]

Note The **both** keyword configures the bidirectional SXP configuration.

Step 3 Exit global configuration mode:

switch(config)# **exit**

Step 4 (Optional) Displays CTS-SXP status and connections:

switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring Bidirectional SXP Support

The following running configuration shows how to configure bidirectional SXP support. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <3.3.3.2> source <3.3.3.1> password <none> mode both vrf <vrf-name>
Warning: The peer should also be configured as both when this peer is configured as both.
```

The following example shows how to display bidirectional SXP configuration details.

```
switch(config)# show run | include connection
cts sxp connection peer 3.3.3.2 source 3.3.3.1 password none mode both vrf management
```

The following example shows the SXP learnt SGT bindings:

```
switch(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x00000000
IP-SGT Mappings as follows:
Total number of IP-SGT Mappings: 0
```

Verifying Cisco TrustSec with SXPv4

The following table provides information about how to verify SXPv4 configuration details.

Commands	Purpose
show cts sxp sgt-map vrf *vrf-name*	Displays information about SXP connection.
show cts sxp connection	Displays detailed information about SXP connections.
show cts sxp connection detail	Displays SXP connection for the specified VRF.
show cts sxp connection vrf *vrf-name*	Displays IP address to SGT mapping.
show cts sxp sgt-map	Displays SXP learnt SGT bindings in detail.
show cts sxp sgt-map detail	Displays the SGT mapping for the specified VRF.

Configuring Static IP-SGT Bindings

You can define a static binding between an IP host address to a security group tag (SGT). The static IP-SGT bindings are configured in the context of a management VRF.

Note Any Cisco TrustSec configuration must be done only in the management VRF.

Before you begin

- Log in to the CLI in EXEC mode.

- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts role-based sgt-map** *ip-address*| *ip-address/IPv4_Length_Prefixsgt_value*	Configures the static binding between an IP host address to a security group tag (SGT). • *ip-address*—IP address of the host. • *ip-address/IPv4_Length_Prefix*—IP address and subnet prefix of the host. • *sgt*—SGT corresponding to the IP address. The range is from 1 to 65519.
Step 3	(Optional) switch(config)# **vrf context**	Specifies the IP-SGT bindings in a VRF context. The default is the default VRF.
Step 4	(Optional) switch(config)# **show cts role-based sgt-map**	Displays the mapping of the IP address to SGT for Cisco TrustSec.
Step 5	(Optional) switch(config)# **show cts ipsgt entries**	Displays all the IP-SGT and Subnet-SGT bindings.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure static IP-SGT bindings:

```
switch# configure terminal
switch(config)# cts role-based sgt-map 1.1.1.1 100
switch(config)# vrf context management
switch(config-vrf)# cts role-based sgt-map 2.2.2.3 200
switch(config-vrf)# exit
switch(config)# show cts role-based sgt-map

IP ADDRESS SGT VRF/VLAN SGT CONFIGURATION
193.191.0.174 2 vlan:971 Device Tracking
193.191.0.176 2 vlan:971 Device Tracking
193.191.0.180 2 vlan:971 Device Tracking
193.191.0.178 2 vlan:971 Device Tracking
25.0.0.4 4 vlan:972 Device Tracking
25.0.0.3 4 vlan:972 Device Tracking
1.1.1.241 411 management CLI Configured
1.1.1.242 421 management CLI Configured
```

```
1.1.1.243 431 management CLI Configured
1.1.1.244 441 management CLI Configured
1.1.1.245 451 management CLI Configured
1.1.2.49  491 management CLI Configured
1.1.2.50  501 management CLI Configured
1.1.2.51  511 management CLI Configured
1.1.12.46 461 management CLI Configured
1.1.12.47 471 management CLI Configured
1.12.1.48 481 management CLI Configured
2.2.2.2   3   management CLI Configured
25.0.0.3 4 management SXP peer:10.197.130.185
25.0.0.4 4 management SXP peer:10.197.130.185
25.0.0.5 5 management SXP peer:10.197.130.185

switch(config)# show cts ipsgt entries vrf management

Interface SGT IP ADDRESS VRF/VLAN Learnt
-------------- ------ ------------ ---------- ---------

- 3 2.2.2.2 management CLI Configured
- 4 25.0.0.3 management SXP peer: 10.197.130.185
- 4 25.0.0.4 management SXP peer: 10.197.130.185
- 5 25.0.0.5 management SXP peer: 10.197.130.185
```

Note IP-SGT binding can be configured using the CLI or from SXP. Any SGT mapping that is configured from the CLI displays “CLI Configured” under the SGT_CONFIGURATION column.

This example shows how to configure static subnet IP-SGT bindings:

```
switch# configure terminal
switch(config)# cts role-based sgt-map 1.1.1.1 100
switch(config)# vrf context management
switch(config-vrf)#  cts role-based sgt-map 200.200.200.0/24 2000
switch(config-vrf)# exit
switch(config)# show cts role-based sgt-map

IP ADDRESS SGT VRF/VLAN SGT CONFIGURATION
.........................
.........................
200.200.200.0/24    2000          management     CLI Configured
```

Changing the SXP Retry Period

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

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.

- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp retry-period** *seconds*	Specifies the SXP retry timer period. The default value is 60 seconds (1 minute). The range is from 0 to 64000.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the SXP retry period:

```
switch# configure terminal
switch(config)# cts sxp retry-period 60
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
switch(config)#
```

Changing the Interface Delete Hold Timer

The interface delete hold timer period determines how long the interface holds on to the IP-SGT mapping once the interface goes to a nonparticipating state. After the timer expires, the IP-SGT mappings are deleted from the interface and the peers.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts interface delete-hold** *seconds*	Specifies the delete hold timer period for an interface. The default value is 60 seconds (1 minute). The range is from 0 to 64000. If the timer is set to 0, the IP-SGT mappings are deleted instantly. The **no** form of this command does not start the timer when the interface goes to a nonparticipating state and the IP-SGT entries are then always held on the interface.
Step 3	(Optional) switch(config)# **show cts interface delete-hold timer**	Displays the interface delete hold timer period.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the interface delete hold timer:

```
switch# configure terminal
switch(config)# cts interface delete-hold 60
switch(config)# show cts interface delete-hold timer
60
switch(config)#
```

Configuring AAA on the Cisco TrustSec Cisco NX-OS Devices

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

Before you begin

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

Procedure

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

	Command or Action	Purpose
Step 2	**radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **7**] **key** *key-value* **pac authentication accounting**	Configures a RADIUS server host with a key and PAC. The *hostname* argument is alphanumeric, case sensitive, and has a maximum length of 256 characters. The *key* argument is alphanumeric, case sensitive, and has a maximum length of 63 characters. The **0** option indicates that the key is in clear text. The **7** option indicates that the key is encrypted. The default is clear text.
Step 3	(Optional) **show radius-server**	Displays the RADIUS server configuration.
Step 4	**aaa group server radius** *group-name*	Specifies the RADIUS server group and enters RADIUS server group configuration mode.
Step 5	**server** {*ipv4-address* | *hostname*}	Specifies the RADIUS server host address.
Step 6	**use-vrf** *vrf-name*	Specifies the management VRF instance for the AAA server group. **Note** If you use the management VRF instance, no further configuration is necessary for the devices in the network cloud. If you use a different VRF instance, you must configure the devices with that VRF instance.
Step 7	**exit**	Exits RADIUS server group configuration mode.
Step 8	**aaa authorization cts default group** *group-name*	Specifies the RADIUS server groups to use for Cisco TrustSec authorization.
Step 9	**exit**	Exits global configuration mode.
Step 10	(Optional) **show radius-server groups** [*group-name*]	Displays the RADIUS server group configuration.
Step 11	(Optional) **show aaa authorization**	Displays the AAA authorization configuration.
Step 12	(Optional) **show cts pacs**	Displays the Cisco TrustSec PAC information.
Step 13	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure AAA on the Cisco TrustSec Cisco NX-OS devices:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 key L1a0K2s9 pac authentication accounting
switch(config)# aaa group server radius Rad1
switch(config-radius)# server 10.10.1.1
switch(config-radius)# use-vrf management
switch(config-radius)# exit
switch(config)# aaa authentication cts default group Rad1
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Cisco TrustSec Authentication in Manual Mode

You can manually configure Cisco TrustSec on a port profile if your Cisco NX-OS device does not have access to a Cisco Secure ACS. You must manually configure the port profiles on both ends of the connection.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created. **Note** To configure Cisco TrustSec SGTs on an interface, enter the interface configuration mode and specify the interface.
Step 3	switch(config-port-prof)# **cts manual**	Enters Cisco TrustSec manual configuration mode. **Note** You cannot enable Cisco TrustSec on interfaces in half-duplex mode.
Step 4	(Optional) switch(config-port-prof-cts-manual)# **policy dynamic identity** *peer-name*	Configures a dynamic authorization policy download. The *peer-name* argument is the Cisco TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Cisco TrustSec credentials and AAA for Cisco TrustSec.

	Command or Action	Purpose
		Note The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 5	(Optional) switch(config-port-prof-cts-manual)# **policy static sgt** *tag* [**trusted**]	Configures a static authorization policy. The *tag* argument is a hexadecimal value in the format **0x***hhhh*. The range is from 0x2 to 0xffef. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 6	switch(config-port-prof-cts-manual)# **propagate-sgt**	Enables security group tag (SGT) propagation on Layer 2 Cisco TrustSec interfaces.
Step 7	switch(config-port-prof-cts-manual)# **exit**	Exits the current configuration mode.
Step 8	switch(config-port-prof)# **exit**	Exits the current configuration mode.
Step 9	(Optional) switch(config)# **show cts interface all**	Displays the Cisco TrustSec configuration.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec authentication in CTS manual mode:

```
switch# configure terminal
switch(config)# port-profile pp1
switch(config-port-prof)# cts manual
switch(config-port-prof-cts-manual)# policy dynamic identity MyDevice2
switch(config-port-prof-cts-manual)# propagate-sgt
switch(config-port-prof-cts-manual)# exit
switch(config-port-prof)# exit
switch(config)# copy running-config startup-config
```

Configuring SGACL Policies

Manually Configuring SGACL Policies

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

Before you begin

Ensure that you have enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.
Step 2	**cts role-based access-list** *list-name*	Specifies an SGACL and enters role-based access list configuration mode. The *list-name* argument value is alphanumeric, case sensitive, and has a maximum length of 32 characters.
Step 3	(Optional) {**deny** | **permit**} **all**	Denies or permits all traffic.
Step 4	(Optional) {**deny** | **permit**} **icmp**	Denies or permits Internet Control Message Protocol (ICMP) traffic.
Step 5	(Optional) {**deny** | **permit**} **igmp**	Denies or permits Internet Group Management Protocol (IGMP) traffic.
Step 6	(Optional) {**deny** | **permit**} **ip**	Denies or permits IP traffic.
Step 7	(Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]	Denies or permits TCP traffic. The default permits all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
Step 8	{**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]	Denies or permits UDP traffic. The default permits all UDP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
Step 9	**exit**	Exits role-based access-list configuration mode.
Step 10	**cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name*	Maps the SGT values to the SGACL. The *sgt-value* and *dgt-value* argument values range from 0 to 65519. **Note** You must create the SGACL before you can map SGTs to it.

	Command or Action	Purpose
Step 11	(Optional) **show cts role-based access-list**	Displays the Cisco TrustSec SGACL configuration.
Step 12	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure an SGACL policy:

```
switch# configure terminal
switch(config)# cts role-based access-list MySGACL
switch(config-rbacl)# deny all log
switch(config-rbacl)# permit icmp
switch(config-rbacl)# deny igmp
switch(config-rbacl)# permit ip
switch(config-rbacl)# deny tcp dst eq 100
switch(config-rbacl)# permit udp src eq 1312
switch(config-rbacl)# exit
switch(config)# cts role-based sgt 3 dgt 10 access-list MySGACL
witch(config)# copy running-config startup-config
```

Enabling SGACL Policy Enforcement

If you use SGACLs, you must enable SGACL policy enforcement on the port profiles or interfaces that have Cisco TrustSec enabled.

Before you begin

- Ensure that you enabled Cisco TrustSec.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created. **Note** To configure Cisco TrustSec SGTs on an interface, enter the interface configuration mode and specify the interface.
Step 3	switch(config-port-prof)# **cts manual**	Enters CTS manual configuration mode.
Step 4	switch(config-port-prof-cts-manual)# **role-based enforcement**	Enables Cisco TrustSec SGACL policy enforcement on the port profile.

	Command or Action	Purpose
Step 5	switch(config-port-prof-cts-manual)# **exit**	Saves the configuration and exits the current configuration mode.
Step 6	switch(config-port-prof)# **exit**	Saves the configuration and exits the current configuration mode.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable role-based enforcement on a port profile:

```
switch# configure terminal
switch(config-port-prof)# cts manual
switch(config-port-prof-cts-manual)# role-based enforcement
switch(config-port-prof-cts-manual)# exit
switch(config-port-prof)# exit
switch(config)# copy running-config startup-config
```

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 ISE. The Cisco NX-OS software downloads the SGACL policies when it learns of a new SGT through authentication and authorization on an interface.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

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

Refreshing the Downloaded SGACL Policies

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

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

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

Clearing Cisco TrustSec SGACL Policies

You can clear the Cisco TrustSec SGACL policies.

Note The way policies are cleared depends on whether the SGT is static or dynamic. For a static SGT, the SGT is reset to 0 after the flap occurs. For a dynamic SGT, the SGT is downloaded again from the RADIUS server after the flap occurs.

Before you begin

Ensure that you enabled Cisco TrustSec.

Procedure

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

Enabling Statistics for RBACL

You can request a count of the number of packets that match role-based access control list (RBACL) policies. These statistics are collected per ACE.

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

Before you begin

Ensure that you have enabled Cisco TrustSec.

If you plan to enable RBACL statistics, ensure that you have enabled RBACL policy enforcement on the port profile or interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts role-based counters enable**	Enables or disables RBACL statistics. The default is disabled.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show cts role-based counters**	Displays the configuration status of RBACL statistics and lists statistics for all RBACL policies.
Step 6	(Optional) switch# **clear cts role-based counters**	Clears the RBACL statistics so that all counters are reset to 0.

Example

This example shows how to enable statistics for RBACL:

```
switch# configure terminal
switch(config)# cts role-based counters enable
switch(config)# copy running-config startup-config
```

Configuring RBACL Logging

RBACL Logging

You can use role-based access control list (RBACL) logging to monitor flows that affect specific RBACLs. The RBACLs can be configured with the optional log keyword in each of the access control entries (ACEs). When you configure an option, statistics for each flow that match the RBACL permit or deny conditions that you enter are logged in the software. RBACL logging supports both IPv4 and IPv6 addresses.

This example shows how to apply the log option:

```
switch(config)# cts role-based access-list [name]
switch(config-rbacl)#  permit tcp dst gt 1111 log
```

You can enable logging per rule(s) within the RBACL. An implicit deny rule is the default action for RBACLs. To log any packets that match the implicit deny rule, you must create an explicit deny rule and add the **log** keyword.

Statistics and logging are provided for each flow. A flow has the following fields:

- Virtual Supervisor Module (VSM) ID
- Virtual Ethernet Module (VEM) ID
- Security Group Tag (SGT)
- Destination Group Tag (DGT)
- Source IP address
- Source port
- Destination IP address
- Destination port
- Source Interface
- Protocol
- Hit Count

Scalability is provided through the following functionality:

- Each Cisco Nexus 1000V switch can support up to 256 VEMs.
- Each VEM can support up to 5000 permit and 5000 deny flows. The maximum number of permit/deny flows is a configurable option.
- The flow reporting interval can be from 5 to 86,400 seconds (1 day).
- The configuration flow syslog level can be from 0 to 7.
- Up to three syslog servers are supported.

RBACL Flows

An RBACL flow as it pertains to RBACL logging has the following characteristics:

- It represents a stream of IPv4/IPv6 packets with the same packet headers (SrcIP, DstIP, Protocol, SrcPort, DstPort) for which an identical RBACL action is enforced. Each flow entry tracks the count of packets that match the flow.
- It is created only if logging is enabled on the corresponding ingress/egress RBACL policy. Ingress and egress flows are tracked separately.
- Each VEM tracks a maximum of 10,000 ACL flows; a flow space is shared between permit/deny flows, and each has a configurable maximum of 5000.
- Each flow entry contains the following:
  - Packet tuple
  - RBACL action

- Direction
- Packet count

- The RBACL flow lifecycle is as follows:
  - A flow is created when the first packet of a unidirectional stream matches a Layer 3 RBACL policy. A new flow notification is sent to the syslog server.
  - For all subsequent packets with a tuple that matches the flow tuple, the per-flow packet counter is incremented.
  - Each flow is tracked periodically based on the configured reporting interval. Within each periodic report, all the active flows and the corresponding packet count seen since the last periodic report are reported to the syslog server.
  - If no packets match a flow for one full periodic interval, the flow entry is purged. This process is the only flow-aging scheme.
  - A flow is not stateful. There is no connection tracking for TCP flows.

- The flow reporting process occurs in the following manner:
  - For each flow created, a new flow notification message is sent to the syslog server.
  - A periodic report for each active flow comes next. A flow is active if packets that match the flow are seen since the last periodic report.
  - The flow information is exported to the syslog server and contains the following: packet tuple, RBACL action, direction, VEM ID, VSM ID, packet count.
  - The periodic time can be as low as 5 seconds with the default setting of 5 minutes. A new user space RBACL-logging thread handles the periodic poll and report functionality.
  - Syslog messages that identify the flow space usage are sent at 75 percent, 90 percent, and 100 percent of the threshold maximum to the syslog server once during each interval.

Syslog Messages

Syslog message characteristics are as follows:

- Syslog messages that contain flow information are exported from each Virtual Ethernet Module (VEM).
- The syslog client functionality is RFC-5424 compliant and communicates to servers over a UDP port (514).
- The host must be configured with a vmknic interface that can reach the remote syslog server.
- On an ESXi-5.0 host, syslog messages are blocked by a firewall. The Cisco Nexus 1000V has installation scripts that open the firewall for port 514.

Configuring RBACL Logging

By default, RBACL logging is enabled on all Virtual Ethernet Modules (VEMs). In addition, the following rules apply to RBACL logging configuration:

- Any rule can be enabled for logging by adding the log keyword.
- Only packets that have a rule with the log keyword enabled are logged.

Disabling RBACL Logging

You can disable RBACL logging on a VEM by entering the following command:

Command	Purpose
[**no**] **logging ip access-list cache module** *vem*	Disables RBACL logging on the specified VEM.

Configuring a Time Interval for Accumulating Packet Counters

You can configure the time interval for accumulating packet counters before they are reported to the syslog servers. You enter the time range in seconds from 5 to 86,400 seconds (1 day). The default is 300 seconds (5 minutes).

You can configure the amount of time to accumulate packet counters by entering one of the following commands:

Command	Purpose
logging ip access-list cache interval *secs*	Sets the time interval in seconds to accumulate packet counters before they are reported to the syslog servers, where *secs* is the number of seconds.
[**no**] **logging ip access-list cache interval** *secs*	Reverts the configuration to the default time interval configuration 300 seconds (5 minutes), where *secs* is the number of seconds.

This example shows the time interval syslog message format that is sent periodically when the time interval expires:

```
Oct 6 16:58:54 172.23.180.5 1 1988-01-19T07:11:27.108 172.23.180.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-INTERVAL VSM ID: 172.23.180.168, VEM ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 0.0.0.0, Destination IP: 255.255.255.255
Source Port: 68, Destination Port: 67
Source Interface: Veth8, Protocol: "UDP"(17), Hit-count = 91
```

Configuring Flows

You can configure the number of deny and permit flows per VEM. The range is from 0 to 5000 flows; the default is 3000. A syslog message is sent when the flow is near the maximum threshold. The first message is sent when the number of flows has reached 75 percent of the maximum threshold and the next message is sent when the number of flows has reached 90 percent of the maximum threshold. The last message is sent when the number of flows reaches the maximum threshold of 100 percent.

Configuring Permit Flows

You can configure permit flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-permit-flows *num*	Sets the number of permit flows, where *num* is the number of flows.
[**no**] **logging ip access-list cache max-permit-flows**	Reverts the configuration to the default permit flow value of 3000.

These examples show permit flow syslog messages:

- New flow notification message:

```
Oct 6 17:05:10 192.0.2.199 1 1988-01-19T07:17:43.810 192.0.2.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-CREATE VSM ID: 192.0.2.168, VEM ID:
42205f8e-0959-fbe2-6403-bf6d9f75c384
SGT :25 DGT :25 Source IP: 192.0.2.3, Destination IP: 192.0.2.2
Source Port: 40116, Destination Port: 2048
Source Interface: Veth15, Protocol: "TCP"(6), Hit-count = 19
```

- Periodic flow reporting message:

```
Oct 6 17:06:38 192.0.2.5 1 1988-01-19T07:17:53.809 192.0.2.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-INTERVAL VSM ID: 192.0.2.168, VEM ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 192.0.2.2, Destination IP: 192.0.2.3
Source Port: 2048, Destination Port: 40063
Source Interface: Veth6, Protocol: "TCP"(6), Hit-count = 2100
```

- Threshold crossing alarm messages:

```
-  Oct 6 04:17:22 sfish-231-157.cisco.com 1 2011-08-28T11:14:24 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 75 percent

limit (3969)
-  Oct 6 04:17:26 sfish-231-157.cisco.com 1 2011-08-28T11:14:26 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 90 percent

limit (4969)
-  Oct 6 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Configuring Deny Flows

You can configure deny flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-deny-flows *num*	Sets the number of deny flows, where *num* is the number of flows.
[**no**] **logging ip access-list cache max-deny-flows**	Reverts the configuration to the default deny flow value 3000.

These examples show deny flow syslog messages:

- New flow notification message:

```
Oct 6 17:05:10 192.0.2.199 1 1988-01-19T07:17:43.810 192.0.2.168 n1k-acllog -
ACLLOG-DENY-FLOW-CREATE VSM ID: 192.0.2.168, VEM ID: 42205f8e-0959-fbe2-6403-bf6d9f75c384
```

```
SGT :25 DGT :25 Source IP: 192.0.2.3, Destination IP: 192.0.2.2
Source Port: 40116, Destination Port: 2048
Source Interface: Veth15, Protocol: "TCP"(6), Hit-count = 19
```

- Periodic flow reporting message:

```
Oct 6 17:06:38 192.0.2.5 1 1988-01-19T07:17:53.809 192.0.2.168 n1k-acllog -
ACLLOG-DENY-FLOW-INTERVAL VSM ID: 192.0.2.168, VEM ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 192.0.2.2, Destination IP: 192.0.2.3
Source Port: 2048, Destination Port: 40063
Source Interface: Veth6, Protocol: "TCP"(6), Hit-count = 2100
```

- Threshold crossing alarm messages:

```
-  Oct 6 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 75 percent
limit
(4330)
-  Oct 6 04:18:27 sfish-231-157.cisco.com 1 2011-08-28T11:15:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 90 percent
limit
(4630)
-  Oct 6 04:20:17 sfish-231-157.cisco.com 1 2011-08-28T11:17:20 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Configuring Syslog Server Severity Levels

You can set the severity level of a syslog message and up to three remote servers to which you want the message to be sent using the following commands:

Command	Purpose
[**no**] **acllog match-log-level** *level*	Sets the severity level at which syslog messages are sent, where *level* is the severity code from 0 to 7. The no acllog match-log-level level command reverts the RBACL log level to the default severity level 6.
[**no**] **logging ip access-list cache max-deny-flows** *number*	Sets the maximum number of deny flows to *number* per module. The **no logging ip access-list cache max-deny-flows** *number* sets the maximum number of deny-flows to the default value of 3000.
[**no**] **logging ip access-list cache max-permit-flows** *number*	Set the max-permit-flows to a specified number per module. The **no logging ip access-list cache max-permit-flows** *number* sets the maximum number of permit-flows to the default value of 3000.
logging server *A.B.C.D 0-7*	Specifies the syslog server on which you want to set a severity level, where *A.B.C.D* is the syslog server IP address and 0 to 7 are the severity levels you can choose.

Note For ACL logging to work, the RBACL Logging level should be less than or equal to the Syslog level.

The severity level range is from 0 to 7 (default is 6):

Severity Code	Severity Level	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant condition
6 (Default)	Informational	Informational messages
7	Debug	Debug-level messages

Verifying the Cisco TrustSec Configuration

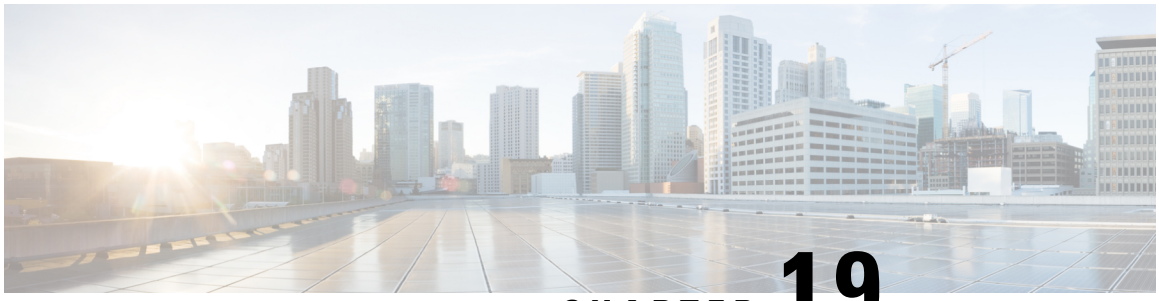
Use the following commands to verify the configuration:

Command	Purpose
show aaa authentication	Displays the AAA authentication configuration on the Cisco Nexus 1000V.
show aaa authorization	Displays the AAA authorization configuration on the Cisco Nexus 1000V.
show cts	Displays the global Cisco TrustSec configuration on the Cisco Nexus 1000V.
show cts sxp	Displays the Cisco TrustSec SXP configuration.
show cts device tracking	Displays the Cisco TrustSec device tracking configuration.
show cts sxp connection	Displays Cisco TrustSec SXP connections.
show cts role-based sgt-map	Displays all the mapping between IP address/Subnet and SGT for Cisco TrustSec.
show cts ipsgt entries	Displays all the IP address/Subnet to SGT mappings.
show cts interface delete-hold timer	Displays the Cisco TrustSec interface delete hold timer period.

Command	Purpose
show cts environment-data	Displays Cisco TrustSec environmental data.
show cts interface	Displays the Cisco TrustSec configuration for all interfaces.
show cts interface ethernet *slot*/*port*	Displays the Cisco TrustSec configuration for the specified Ethernet interface.
show cts interface vethernet *number*	Displays the Cisco TrustSec configuration for the specified virtual Ethernet interface.
show cts role-based access-list	Displays Cisco TrustSec SGACL information.
show cts role-based counters	Displays the configuration status of RBACL statistics and lists statistics for all RBACL policies.
show cts role-based policy	Displays Cisco TrustSec SGACL policy information.
show running-configuration cts	Displays the running configuration information for Cisco TrustSec.

Feature History for Cisco TrustSec

Feature Name	Feature Name	Releases
Cisco TrustSec Subnet-SGT Mapping	5.2(1)SV3(2.1)	This feature was introduced.
SXPv3 Protocol support	5.2(1)SV3(2.1)	This feature was introduced.
SXP Peer Connections	5.2(1)SV3(1.3)	The peer device can be configured in speaker mode.
Cisco TrustSec	4.2(1)SV2(1.1)	This feature was introduced.



CHAPTER 19

Configuring Traffic Storm Control

This chapter contains the following sections:

Information About Traffic Storm Control

A traffic storm occurs when multicast, broadcast, or unknown-unicast packets flood a port, creating excessive traffic and degrading network performance. Even if the packet rate is not high, the number of clones could be large enough to impact the CPU performance or the switches, servers, and other VEMs on the network. Due to this high CPU usage, the VEM is unable to process the control traffic and traffic disconnects from the VSM. You can use the traffic storm control feature to prevent disruptions from a broadcast, multicast, or unknown-unicast traffic storm on these ports.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a configurable polling interval (the default is 1 second). During this interval, the traffic level 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 in the next polling interval until the traffic decreases below the allowed rate. Storm control works on each traffic type separately, as you can see in these 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 in the next interval. If the broadcast rate is still above the threshold at the start of the next time interval, traffic storm control continues to drop the broadcast traffic.
- 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 in the next interval. If the broadcast and

multicast rate are still above the threshold at the start of the next time interval, traffic storm control continues to drop the broadcast and multicast traffic.

- 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 multicast traffic in the next interval. If the multicast rate is still above the threshold at the start of the next time interval, traffic storm control continues to drop multicast traffic.

Traffic storm control is configurable on every port, either through a port profile or directly on the interface (interface override). On physical interfaces and port channels, you can set the threshold as a percentage of the total available bandwidth, the number of bits per second, or the number of packets per second that the controlled traffic can use. On virtual interfaces, you can set the threshold as the number of bits per second or the number of packets per second 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.

Guidelines and Limitations for Traffic Storm Control

When configuring the traffic storm control level, note the following guidelines and limitations:

- You can configure traffic storm control on a port profile or on an interface (interface override).
- Storm control cannot be configured on member ports of a port channel.
- On physical interfaces, you can set the level as a percentage of the total available bandwidth, number of bits per second, or number of packets per second that the controlled traffic can use.
- On virtual interfaces, you can set the level as a number of bits per second or packets per second.

Default Settings for Traffic Storm Control

This table lists the default settings for traffic storm control parameters.

Table 8: Default Traffic Storm Control Parameters

Parameters	Default
Polling interval	1 second
Traffic storm control	Disabled

Enabling the Traffic Storm Control Feature

Before you can use the Traffic Storm Control feature, you must enable it.

Procedure

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

	Command or Action	Purpose
Step 2	(Optional) **storm-control enable**	Enables the Traffic Storm Control feature.
Step 3	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to enable the Traffic Storm Control feature:

```
switch# configure terminal
switch(config)# storm-control enable
switch(config)# copy running-config startup-config
```

Setting the Traffic Storm Control Polling Interval

The default traffic storm control polling interval is 1 second. You can change this interval using this procedure.

Before you begin

The Traffic Storm Control feature must be enabled. See Enabling the Traffic Storm Control Feature, on page 270.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	(Optional) **storm-control polling-interval** *seconds*	Configures the polling interval as a number of seconds. The range is from 1 to 300 seconds.
Step 3	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure a storm-control level of 40 packets per second on unicast traffic:

```
switch# configure terminal
switch(config)# storm-control polling-interval 2
switch(config)# copy running-config startup-config
```

Configuring Traffic Storm Control on an Ethernet Port Profile

On Ethernet port profiles, you can set the percentage of total available bandwidth, number of bits per second, or number of packets per second that the controlled traffic can use. You can also configure traffic storm control on individual interfaces. To do so, specify the interface instead of the port profile.

Note Traffic storm control uses a default 1-second interval that can affect the behavior of traffic storm control. However, this interval can be changed, as shown in this procedure.

Before you begin

The Traffic Storm Control feature must be enabled. See Enabling the Traffic Storm Control Feature, on page 270.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **storm-control polling-interval** *seconds*	Configures the polling interval as a number of seconds.
Step 3	switch(config)# **port-profile type ethernet** *name*	Enters port profile configuration mode. **Note** You can also configure traffic storm control on individual interfaces. To do so, specify the interface instead of the port profile in this step. For example, specify **interface ethernet** *slot*/*port*.
Step 4	switch(config-port-prof)# **storm-control** {**broadcast** | **multicast** | **unicast**} [*number* **bps** | *number* **pps** | **percentage** *number*]	Configures traffic storm control for traffic on the port profile. The default state is disabled.
Step 5	switch(config-port-prof)# **exit**	Exits port profile configuration mode.
Step 6	(Optional) switch(config)# **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*}	Displays the traffic storm control configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure a storm-control to 40 bits per second on unicast traffic:

```
switch# configure terminal
switch(config)# port-profile type ethernet ethpp
switch(config-port-prof)# storm-control unicast bbp 40
switch(config-port-prof)# exit
switch(config)# copy running-config startup-config
```

Configuring Traffic Storm Control on a vEthernet Port Profile

On the vEthernet port profile, you can set the number of bits per second or packets per second that the controlled traffic can use. You can also configure traffic storm control on individual interfaces. To do so, specify the interface instead of the port profile.

Note Traffic storm control uses a 1-second interval that can affect the behavior of traffic storm control. However, this interval can be changed, as shown in this procedure.

Before you begin

The Traffic Storm Control feature must be enabled. See Enabling the Traffic Storm Control Feature, on page 270.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **storm-control polling-interval** *seconds*	Configures the polling interval as a number of seconds.
Step 3	switch(config)# **port-profile type vethernet** *name*	Enters port profile configuration mode. **Note** You can also configure traffic storm control on individual interfaces. To do so, specify the interface instead of the port profile in this step. For example, specify **interface vethernet** *number*.
Step 4	switch(config-port-prof)# **storm-control** {**broadcast** | **multicast** | **unicast**} [*number* **bps** | *number* **pps**]	Configures traffic storm control for traffic on the port profile. The default state is disabled.
Step 5	switch(config-port-prof)# **exit**	Exits port profile configuration mode.
Step 6	(Optional) switch(config)# **show running-config interface** {**vethernet** *interface-number*}	Displays the traffic storm control configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure a storm-control level of 40 packets per second on unicast traffic:

```
switch# configure terminal
switch(config)# port-profile type vethernet vethpp
switch(config-port-prof)# storm-control unicast 40 pps
switch(config-if)# exit
switch(config)# copy running-config startup-config
```

Verifying Traffic Storm Control Configuration

To display traffic storm control configuration information, perform one of the following tasks:

Command	Purpose
vemcmd show storm stats	Displays the traffic storm control statistics for the VEM.
vemcmd show storm status	Displays the traffic storm control status of the VEM.
show running-config interface	Displays the traffic storm control configuration.
show running-config port-profile *name*	Displays the traffic storm control configuration of the specified port profile.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 1000V Command Reference*.

Configuration Example for Traffic Storm Control

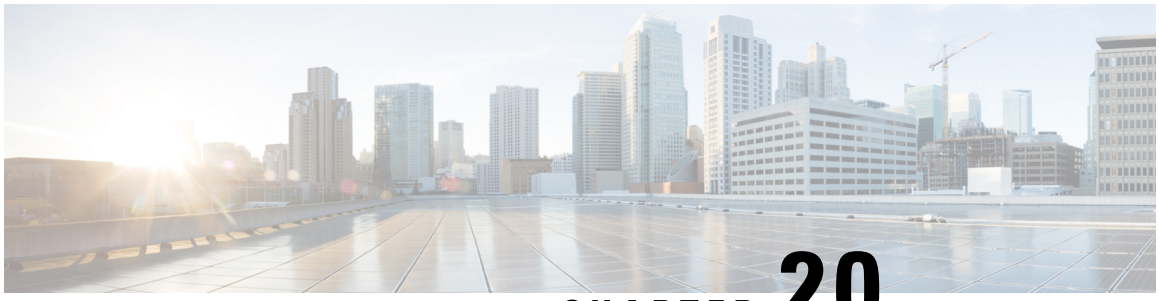
The following example shows how to configure traffic storm control:

```
interface Ethernet1/1
  storm-control broadcast pps 40
  storm-control multicast pps 40
  storm-control unicast pps 40
```

Feature History for Traffic Storm Control

This table only includes updates for those releases that have resulted in additions to the feature.

Feature Name		Feature Information
Traffic Storm Control	5.2(1)SV3(1.1)	This feature was introduced.



CHAPTER 20

Configuring Layer 3 Security

This chapter contains the following sections:

Information About Layer 3 Security

Layer 3 Security (L3Sec) is a framework that secures the internal control plane communications (control and packet traffic) of the Cisco Nexus 1000V in a more robust way than in previous releases. It operates only in Layer 3 Control mode.

When you install a Cisco Nexus 1000V switch with release 5.2(1)SV3(1.1) or higher or when you change the service (svs) mode from Layer 2 to Layer 3 on a switch that is running release 5.2(1)SV3(1.1), the Layer 3 Security (L3sec) feature is enabled by default. However, when you upgrade to release 5.2(1)SV3(1.1), the L3sec setting prior to the upgrade (disabled) is carried over, so the setting is disabled by default. You can enable the L3sec setting manually using the CLI.

Enabling and Disabling the Layer 3 Security Feature

You can enable or disable the Layer 3 security (L3sec) feature.

Before you begin

You are logged in to the CLI in EXEC mode.

Your VSM is configured in Layer 3 control mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**svs-domain**	Places you into SVS domain configuration mode.

	Command or Action	Purpose
Step 3	[**no**] **enable l3sec**	Enables the L3sec feature. Using the **no** option disables the feature.
Step 4	**show running-config**	Displays the l3sec configuration under svs-domain configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

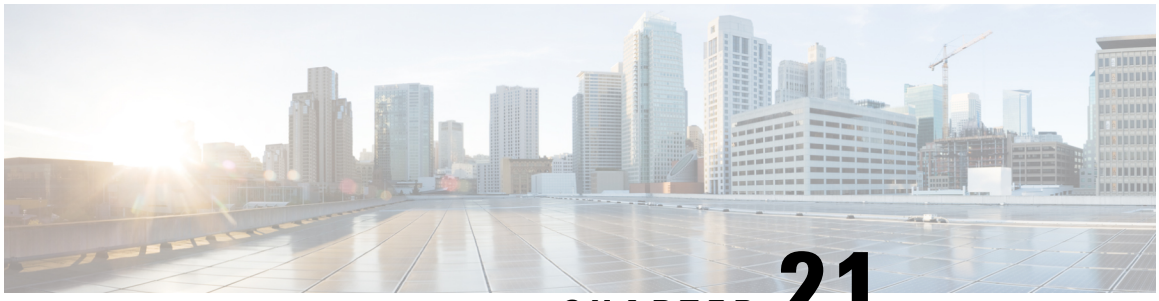
This example shows how to enable the L3sec feature:

```
switch# configure terminal
switch(config)# svs-domain
switch(config-svs-domain)# enable l3sec
switch(config-svs-domain)# copy running-config startup-config
```

Feature History for Layer 3 Security

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
L3sec	5.2(1)SV3(1.1)	This feature was introduced.



CHAPTER 21

Configuring 802.1X

This chapter contains the following sections:

Information About 802.1X

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

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

Device Roles

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

The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and Cisco Nexus 1000v device services and responds to requests from the Cisco Nexus 1000v device. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows operating device.

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

Authentication server
The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the Cisco Nexus 1000v device regarding whether the supplicant is authorized to access the LAN and Cisco Nexus 1000v device services. Because the Cisco Nexus 1000v 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 vEthernet 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 vEthernet and sends to the supplicant.

Note The Cisco Nexus 1000v device can only be an 802.1X authenticator.

Authentication Initiation and Message Exchange

Either the authenticator (Cisco Nexus 1000v 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.

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

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

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

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.

Single Host and Multiple Hosts Support

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

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the Cisco Nexus 1000v device puts the port in the authorized state. When the endpoint device leaves the port, the Cisco Nexus 1000v 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 (vEthernet access port) of the Cisco Nexus 1000v device.

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

Licensing Requirements for 802.1X

The following table shows the licensing requirements for this feature:

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

Prerequisites for 802.1X

802.1X Guidelines and Limitations

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

- All the Dot1x configurations are supported only in the port-profile mode.
- Use of **dot1x pae authenticator** command in any form is not recommended. Use of this command might result in undefined behavior in Dot1x state machine. You can use **dot1x port-control auto** command in the port-profile to control Dot1x configuration.
- The Cisco Nexus 1000v software supports 802.1X authentication only on vEthernet ports.
- The Cisco Nexus 1000v software does not support 802.1X authentication on port channels or subinterfaces.
- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an vEthernet interface.
- The Cisco NX-OS software supports 802.1X authentication only on vEthernet interfaces that are in a port channel, a trunk, or an access port.
- The Cisco NX-OS software does not support single host mode on trunk interfaces .
- The Cisco NX-OS software does not support MAC address authentication bypass.

- The Cisco NX-OS software does not support the following 802.1X protocol enhancements:
  - One-to-many logical VLAN name to ID mapping
  - Web authorization
  - Dynamic domain bridge assignment
  - IP telephony
  - Mac authentication bypass
  - 802.1x specific SNMP MIBs
- For RADIUS Accounting, only the start and stop messages with basic attributes such as Username, Network Device Name, Calling Station ID(MAC Address), NAS IP Address (Network device IP address), and AAA Session ID are supported.
- Configuring VSM as SXP speaker with CTS device tracking option populates the ISE server with IP-SGT mapping that can be used instead of the Framed IP address for Radius Accounting.

Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

Table 9: Default 802.1X Parameters

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

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

Configuring 802.1X

This section describes how to configure the 802.1X feature.

Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

Procedure

Step 1 Enable the 802.1X feature.

Step 2 Configure the connection to the remote RADIUS server.

Step 3 Enable 802.1X feature on the vEthernet interfaces.

Enabling the 802.1X Feature

You must enable the 802.1X feature on the Cisco Nexus 1000v device before authenticating any supplicant devices.

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose

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

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status. **Note** The **show dot1x** command is available if the 802.1X feature is enable. You can also use the **show feature** command to verify the status of the 802.1X feature.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Authentication Methods for 802.1X

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Cisco Nexus 1000v device can perform 802.1X authentication.

Before you begin

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

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose

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

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

Controlling 802.1X Authentication on an Interface

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

Auto
Enables 802.1X authentication on the interface.

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

Force-unauthorized
Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v device.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name* **Example:**	Selects the port-profile to configure and enters port-profile configuration mode.

	Command or Action	Purpose
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1` `switch(config-port-prof)#`	
Step 3	**dot1x port-control** {**auto** | **force-authorized** | **forced-unauthorized**} **Example:** `switch(config-port-prof)# dot1x port-control auto`	Changes the 802.1X authentication state on the interface. The default is force-authorized.

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

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

Enabling Periodic Reauthentication for Port-Profile

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

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v device.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`Switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

Step 3	**dot1x re-authentication** **Example:** `switch(config-port-prof)# dot1x re-authentication`	Enables periodic reauthentication of the supplicants connected to the virtual interface. By default, periodic authentication is disabled.
Step 4	(Optional) **dot1x timeout re-authperiod** *seconds* **Example:** `switch(config-port-prof)# dot1x timeout re-authperiod 3300`	Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535. **Note** This command affects the behavior of the Cisco Nexus 1000v device only if you enable periodic reauthentication on the virtual interface.

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

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

Manually Reauthenticating Supplicants

You can manually reauthenticate the supplicants for the entire Cisco Nexus 1000v device or for a virtual interface.

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v device.

Procedure

	Command or Action	Purpose
Step 1	**dot1x re-authenticate** [**vethernet** *port*] **Example:** `switch# dot1x re-authenticate vethernet 1`	Reauthenticates the supplicants on the Cisco Nexus 1000v device or on a virtual interface.

Manually Initializing 802.1X Authentication

You can manually initialize the authentication for all supplicants on a Cisco Nexus 1000v device or for a specific interface.

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v device.

Procedure

	Command or Action	Purpose
Step 1	**dot1x initialize** [**interface vethernet** *port*] **Example:** `switch# dot1x initialize interface vethernet 1`	Initializes 802.1X authentication on the Cisco Nexus 1000v device or on a specified interface.

Changing 802.1X Authentication Timers for a Port-Profile

You can change the following 802.1X authentication timers on the Cisco Nexus 1000v switch interfaces:

Quiet-period timer
When the Cisco Nexus 1000v switch 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.

Switch-to-authentication-server retransmission timer for Layer 4 packets
The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Cisco Nexus 1000v switch waits a set period of time and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP response frames
The supplicant responds to the EAP-request/identity frame from the Cisco Nexus 1000v switch with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP request frames
The supplicant notifies the Cisco Nexus 1000v switch it that received the EAP request frame. If the authenticator does not receive this notification, it waits a set period of time and then retransmits the frame. The default is the value of the global retransmission period timer. The range is from 1 to 65535 seconds.

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

Step 3	(Optional) **dot1x timeout quiet-period** *seconds* **Example:** `switch(config-port-prof)# dot1x timeout quiet-period 25`	Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 4	(Optional) **dot1x timeout ratelimit-period** *seconds*	Sets the number of seconds that the authenticator ignores EAPOL-Start packets

	Command or Action	Purpose
	Example: `switch(config-port-prof)# dot1x timeout ratelimit-period 10`	from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.
Step 5	(Optional) **dot1x timeout server-timeout** *seconds* **Example:** `switch(config-port-prof)# dot1x timeout server-timeout 60`	Sets the number of seconds that the Cisco Nexus 1000v switch waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 6	(Optional) **dot1x timeout supp-timeout** *seconds* **Example:** `switch(config-port-prof)# dot1x timeout supp-timeout 20`	Sets the number of seconds that the Cisco Nexus 1000v switch waits for the supplicant to respond to an EAP request frame before the Cisco Nexus 1000v switch retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 7	(Optional) **dot1x timeout tx-period** *seconds* **Example:** `switch(config-port-prof)# dot1x timeout tx-period 40`	Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

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

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

Enabling Single Host or Multiple Hosts Mode

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

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

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

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

Enabling 802.1x Guest VLAN

Guest VLAN configuration is used to provide limited network accessibility to a VM user when the VM does not have 802.1x capability or when the VSM is not available (Headless mode).

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

Step 3	**authentication event no-response action authorize vlan** *vlan-id* **Example:** `switch(config-port-prof)#authentication event no-response action authorize vlan 1309`	Configures and enables a guest VLAN on a particular port-profile. **Note** To disable the guest VLAN feature on a particular port-profile, use the no form of this command.

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

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

Disabling 802.1X Authentication

You can disable 802.1X authentication on the Cisco Nexus 1000v switch device. By default, the Cisco Nexus 1000v software enables 802.1X authentication after you enable the 802.1X feature. However, when you disable the 802.1X feature, the configuration is removed from the Cisco Nexus 1000v switch. The Cisco Nexus 1000v software allows you to disable 802.1X authentication without losing the 802.1X configuration.

Note When you disable 802.1X authentication, the port mode for all interfaces defaults to force-authorized regardless of the configured port mode. When you reenable 802.1X authentication, the Cisco Nexus 1000v software restores the configured port mode on the interfaces.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**no dot1x system-auth-control** **Example:** `switch(config)# no dot1x system-auth-control`	Disables 802.1X authentication on the Cisco Nexus 1000v switch. The default is enabled. **Note** Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Cisco Nexus 1000v switch.

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

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

Disabling the 802.1X Feature

You can disable the 802.1X feature on the Cisco Nexus 1000v switch.

When you disable 802.1X, all related configurations are automatically discarded. The Cisco Nexus 1000v software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Cisco Nexus 1000V for VMware vSphere System Management Configuration Guide, Release 5.x* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

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

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

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

Resetting the 802.1X Port-Profile Configuration to the Default Values

You can reset the 802.1X configuration for a virtual interface to the default values.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

Step 3	**dot1x default** **Example:** `switch(config-port-prof)# dot1x default`	Reverts to the 802.1X configuration default values for the virtual interface.

	Command or Action	Purpose

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

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

Setting the Maximum Authenticator-to-Supplicant Frame Retransmission Retry Count for a Port-Profile

You can set the maximum number of times that the Cisco Nexus 1000v switch retransmits authentication requests to the supplicant on a virtual interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

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

	Command or Action	Purpose
		Note Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified virtual interface.

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

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

Enabling RADIUS Accounting for 802.1X Authentication

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

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

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

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

Configuring AAA Accounting Methods for 802.1X

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

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

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

Example

This example shows how to enable the 802.1x feature:

```
switch# configure terminal
switch(config)# aaa accounting dot1x default group radius
switch(config)# exit
```

```
switch# show aaa accounting
switch# copy running-config startup-config
```

Setting the Maximum Reauthentication Retry Count on a Port-Profile

You can set the maximum number of times that the Cisco Nexus 1000v switch retransmits reauthentication requests to the supplicant on a virtual interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose

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

Step 2	**port-profile type vethernet** *port_profile_name*
	Example:
	`switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1`
	`switch(config-port-prof)#`
		Selects the port-profile to configure and enters port-profile configuration mode.

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

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

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

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 vethernet *port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an vEthernet 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 1000V for VMware vSphere Command Reference, Release 5.x* for your platform.

Monitoring 802.1X

You can display the statistics that the Cisco Nexus 1000v switch maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Cisco Nexus 1000v switch.

Procedure

	Command or Action	Purpose
Step 1	**show dot1x** {**all** | **interface vethernet** *port*} **statistics** **Example:** `switch(config-port-prof)# show dot1x all statistics`	Displays the 802.1X statistics.

Configuration Example for 802.1X

The following example shows how to configure 802.1X on the port-profile for a port-profile:

```
configure terminal
feature dot1x
aaa authentication dot1x default group SAMPLE_RADIUS_USERS_GROUP_1
port-profile type vethernet SAMPLE_PORT_PROFILE_1
```

```
dot1x port-control auto
```

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

802.1X integration with Cisco Trustsec

With this release, 802.1X can function with Cisco Trustsec (CTS) feature. For detailed information about Cisco Trustsec, see Configuring Cisco Trustsec. You need advanced license for Nexus 1000v to enable CTS feature. When you configure CTS with 802.1X:

- If dot1x port-control is configured together with CTS, dot1x SGT is obtained from radius server and it takes priority.
- Ensure that the **cts manual** command is configured before configuring the **dot1x port-control auto** command while configuring port-profile for CTS.

The following is a sample configuration to integrate 802.1X feature with CTS.

```
Before enabling 802.1X and CTS commands on a port-profile:
-----------------------------------------------------------
port-profile type vethernet SAMPLE_PORT_PROFILE_1
switchport mode access
switchport access vlan 1309
no shutdown
state enabled
vmware port-group

Enabling 802.1X and CTS on a port-profile:
------------------------------------------
First configure 'cts manual' and then configure 'dot1x port-control auto' as below:
switch# configure terminal
switch(config)# port-profile type vethernet SAMPLE_PORT_PROFILE_1
switch(config-port-prof)# cts manual
switch(config-port-prof-cts-manual)# exit
switch(config-port-prof)# dot1x port-control auto
switch(config-port-prof)# end
switch#
```