Cisco Nexus 1000V for Microsoft Hyper-V Security Configuration Guide, Release 5.x

First Published: May 30, 2013

Last Modified: November 17, 2015

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 **Security Overview** 1

User Accounts 1

Authentication, Authorization, and Accounting 1

RADIUS Security Protocol 2

TACACS+ Security Protocol 2

SSH 2

Telnet 3

Access Control Lists 3

Port Security 3

DHCP Snooping 3

Dynamic ARP Inspection 4

IP Source Guard 4

CHAPTER 2 **Managing User Accounts** 5

Information About User Accounts 5

Role 6

Username 6

Password 6

Check of Password Strength 7

Expiration Date 7

Guidelines for Creating User Accounts 8

Guidelines for Creating a Role 8

Default Settings for User Access 9

Configuring User Access 9

Enabling the Check of Password Strength 9

Disabling the Check of Password Strength 10

Creating a User Account 10

Creating a Role 12

Creating a Feature Group 13
Configuring Interface Access 14
Configuring VLAN Access 15
Configuration Examples 16
Configuration Example for Creating a Feature Group 16
Configuration Example for Creating a Role 17
MIBs 17
Feature History for User Accounts 17

CHAPTER 3 **Configuring Authentication, Authorization, and Accounting 19**
Information About AAA 19
AAA Security Services 19
Authentication 20
Authorization 22
Accounting 22
AAA Server Groups 22
Prerequisites for AAA 22
Guidelines and Limitations 22
AAA Default Settings 23
Configuring AAA 23
Configuring a Login Authentication Method 23
Enabling Login Authentication Failure Messages 24
Verifying the AAA Configuration 25
Configuration Examples for AAA 26
Feature History for AAA 26

CHAPTER 4 **Configuring RADIUS 27**
Information About RADIUS 27
RADIUS Network Environments 27
RADIUS Operation 28
RADIUS Server Monitoring 28
Vendor-Specific Attributes 29
Prerequisites for RADIUS 30
Guidelines and Limitations 30
Default Settings 30

Configuring RADIUS Servers 31
Configuring RADIUS Server Hosts 31
Configuring the Global RADIUS Key 32
Configuring a RADIUS Server Key 33
Configuring RADIUS Server Groups 33
Enabling RADIUS Server Directed Requests 35
Setting the Global Timeout for All RADIUS Servers 36
Configuring a Global Retry Count for All RADIUS Servers 36
Setting the Timeout Interval for a Single RADIUS Server 37
Configuring Retries for a Single RADIUS Server 38
Configuring a RADIUS Accounting Server 39
Configuring a RADIUS Authentication Server 40
Configuring Periodic RADIUS Server Monitoring 41
Configuring the Global Dead-Time Interval 42
Manually Monitoring RADIUS Servers or Groups 42
Verifying the RADIUS Configuration 43
Displaying RADIUS Server Statistics 43
Configuration Example for RADIUS 43
Feature History for RADIUS 44

CHAPTER 5 **Configuring TACACS+ 45**
Information About TACACS+ 45
TACACS+ Operation for User Login 46
Default TACACS+ Server Encryption Type and Preshared Key 46
TACACS+ Server Monitoring 47
Vendor-Specific Attributes 47
Cisco VSA Format 47
Prerequisites for TACACS+ 48
Guidelines and Limitations for TACACS+ 48
Default Settings for TACACS+ 48
Configuring TACACS+ 49
Enabling or Disabling TACACS+ 49
Configuring Shared Keys 50
Configuring a TACACS+ Server Host 51
Configuring a TACACS+ Server Group 52

Enabling TACACS+ Server Directed Requests 54
Setting the TACACS+ Global Timeout Interval 55
Setting a Timeout Interval for an Individual TACACS+ Host 56
Configuring the TCP Port for a TACACS+ Host 56
Configuring Monitoring for a TACACS+ Host 57
Configuring the TACACS+ Global Dead-Time Interval 59
Displaying Statistics for a TACACS+ Host 60
Configuration Example for TACACS+ 60
Feature History for TACACS+ 60

CHAPTER 6 **Configuring SSH** 61
Information About SSH 61
SSH Server 61
SSH Client 61
SSH Server Keys 62
Prerequisites for SSH 62
Guidelines and Limitations for SSH 62
Default Settings 63
Configuring SSH 63
Generating SSH Server Keys 63
Configuring a User Account with a Public Key 64
Configuring an OpenSSH Key 64
Configuring IETF or PEM Keys 65
Starting SSH Sessions 67
Clearing SSH Hosts 67
Disabling the SSH Server 68
Deleting SSH Server Keys 68
Clearing SSH Sessions 70
Verifying the SSH Configuration 70
Configuration Example for SSH 71
Feature History for SSH 71

CHAPTER 7 **Configuring Telnet** 73
Information About the Telnet Server 73
Prerequisites for Telnet 73

Guidelines and Limitations for Telnet 73
Default Setting for Telnet 74
Configuring Telnet 74
Enabling the Telnet Server 74
Starting an IP Telnet Session to a Remote Device 74
Clearing Telnet Sessions 75
Verifying the Telnet Configuration 75
Feature History for Telnet 76

CHAPTER 8 **Configuring IP ACLs 77**
Information About ACLs 77
ACL Types and Applications 78
Order of ACL Application 78
Rules 78
Source and Destination 78
Protocols 78
Implicit Rules 79
Additional Filtering Options 79
Sequence Numbers 80
Statistics 80
Prerequisites for IP ACLs 81
Guidelines and Limitations for IP ACLs 81
Default Settings for IP ACLs 81
Configuring IP ACLs 81
Creating an IP ACL 81
Changing an IP ACL 82
Removing an IP ACL 83
Changing Sequence Numbers in an IP ACL 84
Applying an IP ACL as a Port ACL 85
Adding an IP ACL to a Port Profile 86
Applying an IP ACL to the Management Interface 87
Verifying the IP ACL Configuration 89
Monitoring IP ACLs 89
Configuration Example for IP ACL 89
Feature History for IP ACLs 90

CHAPTER 9 **Configuring a MAC ACL** 91

Information About MAC ACLs 91

Prerequisites for MAC ACLs 91

Guidelines and Limitations for MAC ACLs 91

Default Settings for MAC ACLs 92

Configuring MAC ACLs 92

Creating a MAC ACL 92

Changing a MAC ACL 93

Removing a MAC ACL 94

Changing Sequence Numbers in a MAC ACL 95

Applying a MAC ACL as a Port ACL 96

Adding a MAC ACL to a Port Profile 97

Verifying MAC ACL Configurations 98

Monitoring MAC ACLs 99

Configuration Examples for MAC ACLs 99

Configuration Example for Creating a MAC ACL for any Protocol 99

Feature History for MAC ACLs 100

CHAPTER 10 **Configuring Port Security** 101

Information About Port Security 101

Secure MAC Address Learning 101

Static Method 102

Dynamic Method 102

Sticky Method 102

Dynamic Address Aging 102

Secure MAC Address Maximums 103

Interface Secure MAC Addresses 103

Security Violations and Actions 104

Port Security and Port Types 105

Result of Changing an Access Port to a Trunk Port 105

Result of Changing a Trunk Port to an Access Port 105

Guidelines and Limitations for Port Security 105

Default Settings for Port Security 106

Configuring Port Security 106

Enabling or Disabling Port Security on a Layer 2 Interface 106
Enabling or Disabling Sticky MAC Address Learning 107
Adding a Static Secure MAC Address on an Interface 108
Removing a Static or a Sticky Secure MAC Address from an Interface 110
Removing a Dynamic Secure MAC Address 111
Configuring a Maximum Number of MAC Addresses 112
Configuring an Address Aging Type and Time 113
Configuring a Security Violation Action 115
Recovering Ports Disabled for Port Security Violations 117
Verifying the Port Security Configuration 118
Displaying Secure MAC Addresses 118
Configuration Example for Port Security 118
Feature History for Port Security 119

CHAPTER 11 **Configuring DHCP Snooping** 121
Information About DHCP Snooping 121
DHCP Overview 122
BOOTP Packet Format 124
Trusted and Untrusted Sources 126
DHCP Snooping Binding Database 127
DHCP Snooping Option 82 Data Insertion 127
Licensing Requirements for DHCP Snooping 129
Prerequisites for DHCP Snooping 130
Guidelines and Limitations for DHCP Snooping 130
Default Values for DHCP Settings 130
Configuring DHCP Snooping 131
Minimum DHCP Snooping Configuration 131
Enabling or Disabling the DHCP Feature 131
Enabling or Disabling DHCP Snooping Globally 132
Enabling or Disabling DHCP Snooping on a VLAN 133
Enabling or Disabling DHCP Snooping MAC Address Verification 134
Configuring an Interface as Trusted or Untrusted 135
Configuring the Rate Limit for DHCP Packets 136
Detecting Ports Disabled for DHCP Rate Limit Violation 138
Recovering Ports Disabled for DHCP Rate Limit Violations 139

Clearing the DHCP Snooping Binding Database 140
Clearing All Binding Entries 140
Clearing Binding Entries for an Interface 140
Relaying Switch and Circuit Information in DHCP 141
Adding or Removing a Static IP Entry 142
Verifying the DHCP Snooping Configuration 143
Monitoring DHCP Snooping 143
Configuration Example for DHCP Snooping 143
Configuration Example for Trust Configuration and DHCP Server Placement in the Network 145
Standards 147
Feature History for DHCP Snooping 147

CHAPTER 12 **Configuring Dynamic ARP Inspection** 149
Information About Dynamic ARP Inspection 149
ARP 149
ARP Spoofing Attacks 150
DAI and ARP Spoofing 150
Interface Trust and Network Security 151
Prerequisites for DAI 152
Guidelines and Limitations for DAI 152
Default Settings for DAI 152
Configuring DAI Functionality 153
Configuring a VLAN for DAI 153
Configuring a Trusted vEthernet Interface 154
Resetting a vEthernet Interface to Untrusted 156
Configuring DAI Rate Limits 157
Resetting DAI Rate Limits to Default Values 158
Detecting and Recovering Error-Disabled Interfaces 159
Validating ARP Packets 161
Verifying the DAI Configuration 162
Monitoring DAI 162
Configuration Examples for DAI 163
Enabling DAI on VLAN 1 and Verifying the Configuration 164
Dropping an ARP Request Packet and Logging the Error Message 166

Example of Displaying the Statistics for DAI 166
Standards 166
Feature History for DAI 166

CHAPTER 13 **Configuring IP Source Guard 167**
Information About IP Source Guard 167
Prerequisites for IP Source Guard 168
Guidelines and Limitations for IP Source Guard 168
Default Settings for IP Source Guard 168
Configuring IP Source Guard Functionality 169
Enabling or Disabling IP Source Guard on a Layer 2 Interface 169
Verifying the IP Source Guard Configuration 170
Monitoring IP Source Guard Bindings 170
Configuration Example for IP Source Guard 170
Feature History for IP Source Guard 170

CHAPTER 14 **Disabling HTTP Server 171**
Information About the HTTP Server 171
Guidelines and Limitations for the HTTP Server 171
Default Settings for the HTTP Server 171
Disabling the HTTP Server 172
Verifying the HTTP Configuration 172
Related Documents for the Disabling the HTTP Server 173
Standards 173
Feature History for Disabling the HTTP Server 173

CHAPTER 15 **Blocking Unknown Unicast Flooding 175**
Information About UUFB 175
Guidelines and Limitations for UUFB 175
Default Settings for UUFB 176
Configuring UUFB 176
Blocking Unknown Unicast Flooding Globally on the Switch 176
Configuring an Interface to Allow Unknown Unicast Flooding 177
Configuring a Port Profile to Allow Unknown Unicast Flooding 178
Standards 178

Configuration Example for Blocking Unknown Unicast Packets 179
Feature History for UUFB 179

CHAPTER 1

Security Overview

This chapter contains the following sections:

- User Accounts, page 1
- Authentication, Authorization, and Accounting, page 1
- RADIUS Security Protocol, page 2
- TACACS+ Security Protocol, page 2
- SSH, page 2
- Telnet, page 3
- Access Control Lists, page 3
- Port Security, page 3
- DHCP Snooping, page 3
- Dynamic ARP Inspection, page 4
- IP Source Guard, page 4

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.

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

CHAPTER 2

Managing User Accounts

This chapter contains the following sections:

- Information About User Accounts, page 5
- Guidelines for Creating User Accounts, page 8
- Guidelines for Creating a Role, page 8
- Default Settings for User Access, page 9
- Configuring User Access, page 9
- Configuration Examples, page 16
- MIBs, page 17
- Feature History for User Accounts, page 17

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 1: 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 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.

Guidelines for Creating a Role

- You can configure up to 64 user roles.
- You can configure up to up to 256 rules for each role.
- You can assign a single role to more than one user.
- 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.
- 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.

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

Use this procedure to 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

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)# **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	switch(config)# **show password strength-check**	(Optional) Displays the configuration for checking password strength.

	Command or Action	Purpose
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
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

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)# **no password strength-check**	Disables password-strength checking. The default is enabled.
Step 3	switch(config)# **show password strength-check**	(Optional) Displays the configuration for checking password strength.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
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

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)# **show role**	(Optional) 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: • name—A case-sensitive, alphanumeric character string of up to 28 characters in length. • password—The default password is undefined. ◦ 0 = (the default) Specifies that the password you are entering is in clear 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—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	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
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

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- You can configure up to 64 user roles.
- You can configure up to up to 256 rules for each role.
- You can assign a single role to more than one user.
- 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.
- 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	switch(config-role)# **description** *description-string*	(Optional) 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.	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
	Use the **show role feature-group** command to display a list of feature groups. **Example:** 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	switch(config-role)# **show role**	(Optional) Displays the user role configuration.
Step 7	switch(config-role)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

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

Before You Begin

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

	Command or Action	Purpose
		group-name—A case-sensitive, alphanumeric string of up to 32 characters in length.
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	switch(config-role-featuregrp)# **show role feature-group**	(Optional) Displays the feature group configuration.
Step 6	switch(config-role-featuregrp)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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
```

```
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 you have already created by denying access to all interfaces, and then permitting access to selected interfaces.

Before You Begin

Before beginning this procedure you must have done the following:

- Logged in to the CLI in EXEC mode

- Created one or more user roles. In this procedure, you will be modifying a role 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 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	switch(config-role-interface)# **show role** *role-name*	(Optional) Displays the role configuration.
Step 6	switch(config-role-featuregrp)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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: temp
  Vlan policy: permit (default)
  Interface policy: deny
  Permitted interfaces: Ethernet2/1-4
switch(config-role-featuregrp)# copy running-config startup-config
```

Configuring VLAN Access

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

Before You Begin

Before beginning this procedure, you must:

- Be logged in to the CLI in EXEC mode.
- Have already created one or more user roles. In this procedure, you will be modifying a role 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 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	switch(config-role)# **show role** *role-name*	(Optional) Displays the role configuration. role-name is the name you have assigned to the role your created.
Step 6	switch(config-role)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# role name network-observer
switch(config-role)# vlan policy deny
switch(config-role-vlan)# permit interface ethernet 2/1-4
switch(config-role)# show role name network-observer
role: network-observer
  description: temp
  Vlan policy: permit (default)
  Interface policy: deny
  Permitted interfaces: Ethernet2/1-4
switch(config-role)# copy running-config startup-config
```

Configuration Examples

Configuration Example for Creating a Feature Group

```
switch# config 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

```
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	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 3

Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

- Information About AAA, page 19
- Prerequisites for AAA, page 22
- Guidelines and Limitations, page 22
- AAA Default Settings, page 23
- Configuring AAA, page 23
- Verifying the AAA Configuration, page 25
- Configuration Examples for AAA, page 26
- Feature History for AAA, page 26

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 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

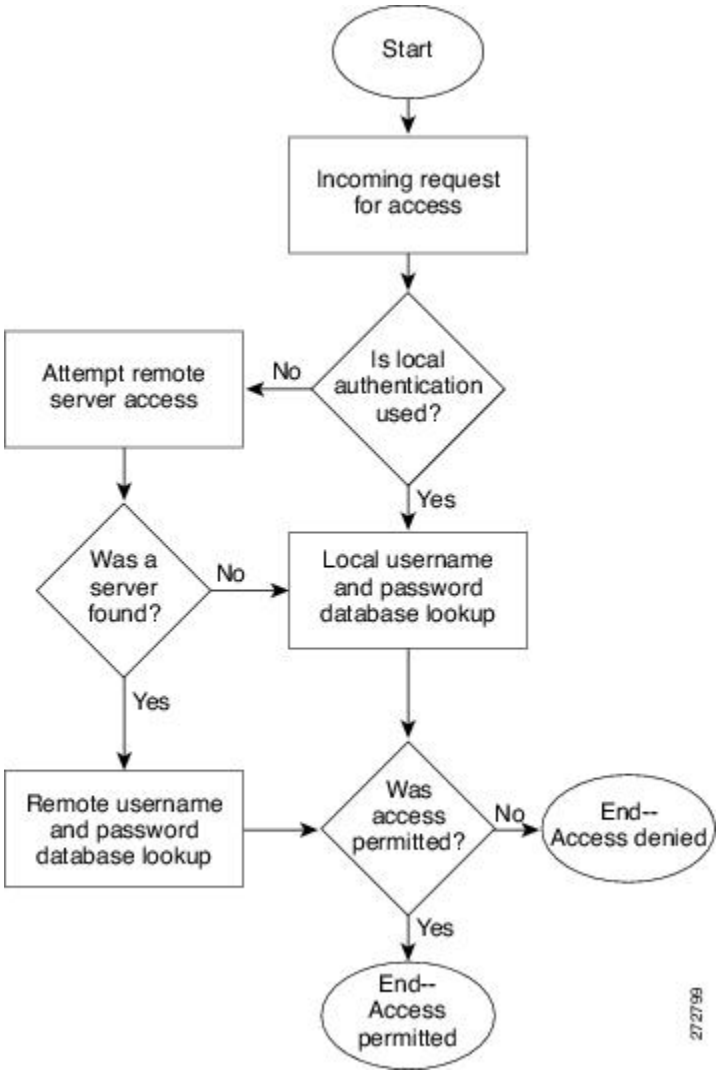
Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and 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 1: 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 have already added the group(s).

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**	Places you into 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—Authentication is done by server group(s) • group-list—List server group names separated by spaces; or none for no authentication. • group-list none— No authentication • local—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—Authentication is done by username.
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.

	Command or Action	Purpose
Step 4	switch# **show aaa authentication**	(Optional) Displays the configured login authentication method.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to 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

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**	Places you into 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	switch# **show aaa authentication login error-enable**	(Optional) Displays the login failure message configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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 one of 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.
show aaa accounting	Displays the AAA accounting configuration.
show aaa authorization	Displays the AAA authorization configuration.
show aaa user default-role	Displays the default role assigned by aaa-admin for remote authentication.
show accounting log	Displays the AAA accounting log.
show encryption service stat	Displays the encryption service status.

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)
```

Example: show aaa accounting

```
switch# show aaa accounting
default: local
```

Example: show aaa authorization

```
switch# show aaa authorization
pki-ssh-cert: local
         pki-ssh-pubkey: local
AAA command authorization:
```

Example: show aaa user default-role

```
switch# show aaa user default-role
enabled
```

Example: show accounting log

```
switch# show accounting log
Thu May 16 14:22:30 2013:type=stop:id=ppm.2748:user=admin:cmd=
Thu May 16 14:22:58 2013:type=start:id=unknown_session:user=root:cmd=
Thu May 16 14:22:58 2013:type=update:id=unknown_session:user=root:cmd=updated v3
 user : admin
Thu May 16 14:22:58 2013:type=update:id=unknown_session:user=root:cmd=configure
terminal ; username admin password ******** role network-admin (SUCCESS)
Thu May 16 14:22:58 2013:type=stop:id=unknown_session:user=root:cmd=
Thu May 16 14:23:07 2013:type=start:id=unknown_session:user=root:cmd=
Thu May 16 14:23:07 2013:type=update:id=unknown_session:user=root:cmd=system red
undancy role standalone (SUCCESS)
```

Example: show encryption service stat

```
switch# show encryption service stat
Encryption service not enabled
Master Encryption Key: not configured.
Type-6 encryption is not being used
```

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	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 4

Configuring RADIUS

This chapter contains the following sections:

- Information About RADIUS, page 27
- Prerequisites for RADIUS, page 30
- Guidelines and Limitations, page 30
- Default Settings, page 30
- Configuring RADIUS Servers, page 31
- Verifying the RADIUS Configuration, page 43
- Displaying RADIUS Server Statistics, page 43
- Configuration Example for RADIUS, page 43
- Feature History for RADIUS, page 44

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 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.

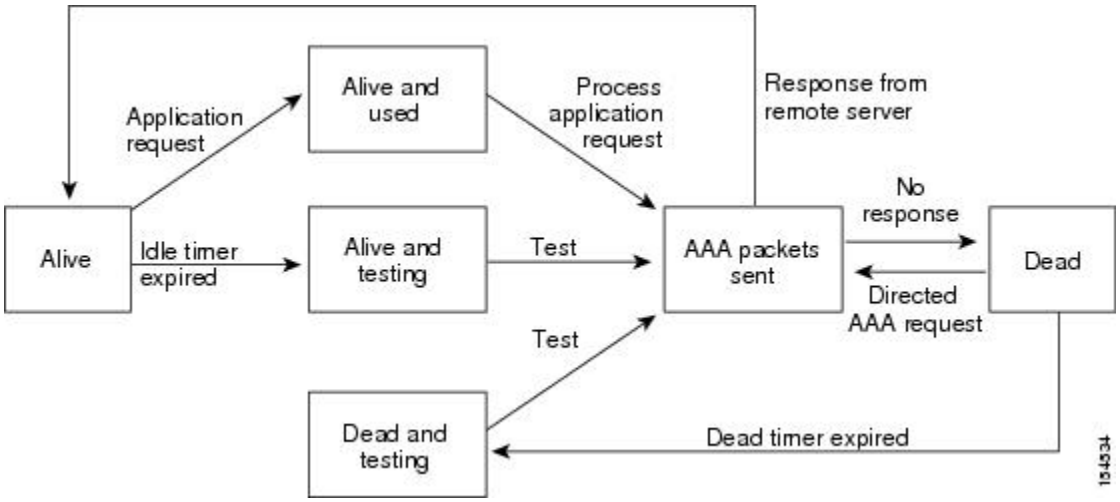
RADIUS Server Monitoring

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 2: 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 and vdc-admin, the value field would be “network-operator vdc-admin.” 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 vdc-admin”
  ```

  ```
  shell:roles*“network-operator vdc-admin”
  ```

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

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

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

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

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

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 2: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting

Parameters	Default
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

Configuring RADIUS Server Hosts

Use this procedure to configure the IP address 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

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**	Places you into global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*}	Defines the IP address or hostname for the RADIUS server, or the RADIUS server Domain Name Server (DNS) name. hostname—alphanumeric, case sensitive, and has a maximum of 256 characters.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration

	Command or Action	Purpose
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to configure the key that is used by all RADIUS servers to authenticate with the Cisco Nexus 1000V.

You must know the global key that is used for RADIUS server authentication.

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

```
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 Server Key

Use this procedure to configure a key for a single RADIUS server host.

You must have the key to be used for the remote RADIUS host

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)# **radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **6** | **7**] *key-value*	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**), encrypted shared secret (**6**), or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the show running-config command to display the encrypted preshared keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Configuring RADIUS Server Groups

Use this procedure to 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

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- 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* | *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	switch(config-radius)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	switch(config-radius)# **use-vrf** *vrf-name*	(Optional) Specifies the VRF to use to contact the servers in the server group
Step 6	switch(config-radius)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) 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
Step 7	switch(config-radius)# **show radius-server groups** [*group-name*]	(Optional) Displays the RADIUS server group configuration.
Step 8	switch(config-radius)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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 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

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

```
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 the Global Timeout for All RADIUS Servers

Use this procedure to 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

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**	Places you into 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 the EXEC mode.
Step 4	switch(config-radius)# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch(config-radius)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
switch# configure terminal
switch(config)# n1000v(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

Use this procedure to 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

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**	Places you into 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 the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# radius-server retransmit 31
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Setting the Timeout Interval for a Single RADIUS Server

Use this procedure to 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

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)# **radius-server host** { *ipv4-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 the EXEC mode.

	Command or Action	Purpose
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to 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

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

You should 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* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. The default 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 the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration

	Command or Action	Purpose
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to configure a server to perform accounting functions.

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

Before You Begin

Before beginning this procedure:

- You must be logged in to the CLI in EXEC mode.
- You should 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	switch(config)# **radius-server host** { *ipv4-address* | *host-name*} **acct-port** *udp-port*	(Optional) 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	switch(config)# **radius-server host** { *ipv4-address* | *host-name*} **accounting**	(Optional) Designates the specific RADIUS host as an accounting server. The default is both accounting and authentication.
Step 4	switch(config)# **exit**	Returns you to the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to configure a server to perform authentication functions.

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

Before You Begin

Before beginning this procedure:

- You must be logged in to the CLI in EXEC mode.
- You should 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	switch(config)# **radius-server host** { *ipv4-address* | *hostname*} **auth-port** *udp-port*	(Optional) 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	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **authentication**	(Optional) Designates the specific RADIUS host as an authentication server. The default is both accounting and authentication.
Step 4	switch(config)# **exit**	Returns you to the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to 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

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)# **radius-server host** {*ipv4-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 the EXEC mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to 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

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

```
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

Use this procedure to manually send a test message to a RADIUS server or to a server group.

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

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

Verifying the RADIUS Configuration

Use one of 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*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

Displaying RADIUS Server Statistics

Use the following command to display statistics for RADIUS sever activity.

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

Configuration Example for RADIUS

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

```
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	5.2(1).SM1(5.1)	This feature was introduced.

CHAPTER 5

Configuring TACACS+

This chapter contains the following sections:

- Information About TACACS+, page 45
- Prerequisites for TACACS+, page 48
- Guidelines and Limitations for TACACS+, page 48
- Default Settings for TACACS+, page 48
- Configuring TACACS+, page 49
- Displaying Statistics for a TACACS+ Host, page 60
- Configuration Example for TACACS+, page 60
- Feature History for TACACS+, page 60

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 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:

   a ACCEPT—User authentication succeeds and service begins. If user authorization is needed, authorization begins.

   b REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.

   c ERROR—An error occurred at some time during authentication either at the daemon or in the network connection. 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

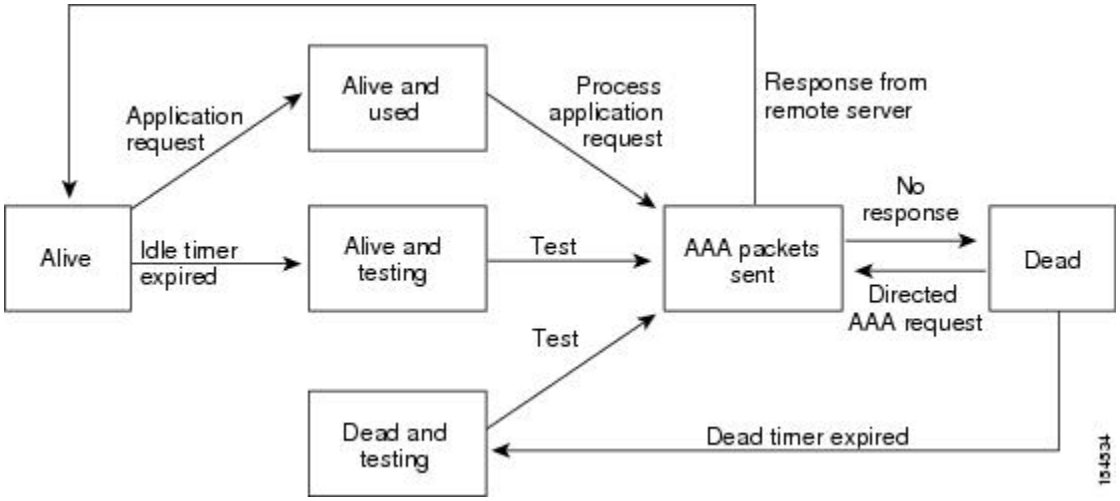
You must configure the TACACS+ preshared key to authenticate to the TACACS+ server. A preshared key is a secret text string shared between the device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not 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 3: TACACS+ Server States



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

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.

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+

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

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**	Places you into global configuration mode.
Step 2	switch(config)# [**no**] **tacacs+ enable**	Enables or disables TACACS+.
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

```
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.

Use this procedure to configure the following:

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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.
Step 2	switch(config)# **tacacs-server key** [**0** | **6** | **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, the default. • 6—Specifies the shared secret • 7—Specifies an encrypted string (key) to follow. • *global_key*: A 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* | *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, the default. 7—Specifies an encrypted string (key) to follow. *global key*—A string of up to 63 characters.

	Command or Action	Purpose
		This shared key is used instead of the global shared key.
Step 4	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 5	switch# **show tacacs-server**	(Optional) 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	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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

Before beginning this procedure, you must have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured 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**	Places you into global configuration mode.

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

```
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

Use this procedure to 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

Before beginning this procedure, you must be sure of the following:

- You are logged in to the CLI in EXEC mode.
- All servers added to a TACACS+ server group use the TACACS+ protocol.
- You have already configured the preshared keys.
- You have already enabled TACACS+ for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**t	Places you into 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* | *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. **Note** If the specified TACACS+ server is not found, configure it using the tacacs-server host command and retry this command.
Step 4	switch(config-tacacs+)# **deadtime** *minutes*	(Optional) 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.
Step 5	switch(config-tacacs+)# **use-vrf** *vrf-name*	(Optional) Specifies the virtual routing and forwarding instance (VRF) to use to contact this server group
Step 6	switch(config-tacacs+)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) 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	switch(config-tacacs+)# **show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration
Step 8	switch(config-tacacs+)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

This procedure allows you to designate the TACACS+ server to send their authentication request to. This is called a directed-request.

When directed requests are enabled, the user can log in as username@vrfname:hostname, where *vrfname* is the VRF to use and hostname is the name of a configured TACACS+ server.

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

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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 the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server directed-request**	(Optional) Displays the TACACS+ directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

```
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

Use this procedure to 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

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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 the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(confi)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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

Use this procedure to 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

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured the TACACS+ server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-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 the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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

Use this procedure to configure a TCP port other than port 49 (the default for TACACS+ requests).

Before You Begin

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured the TACACS+ server

Procedure

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

```
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

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured the TACACS+ server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **tacacs-server host** { *ipv4-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**: The default is test. **Note** To protect network security, we recommend assigning a username that is not already in the TACACS+ database. • **password**: 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 the global configuration mode and returns you to EXEC mode.
Step 5	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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

Use this procedure to 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

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured the TACACS+ server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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 the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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*}

Configuration Example for TACACS+

The following 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+	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 6

Configuring SSH

This chapter contains the following sections:

- Information About SSH, page 61
- Prerequisites for SSH, page 62
- Guidelines and Limitations for SSH, page 62
- Default Settings, page 63
- Configuring SSH, page 63
- Verifying the SSH Configuration, page 70
- Configuration Example for SSH, page 71
- Feature History for SSH, page 71

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.
- 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

Use this procedure to 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

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**	Places you into 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	switch# **show ssh key**	(Optional) Displays the SSH server keys.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configue 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

Use this procedure to specify the SSH public keys in OpenSSH format for user accounts.

Use this procedure to 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

Before beginning this procedure, be sure you have:

- Logged in to the CLI in EXEC mode
- Generated an SSH public key in OpenSSH format
- An existing user account

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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 you to EXEC mode.
Step 4	switch# **show user-account**	(Optional) Displays the user account configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to specify the SSH public keys in IETF SECSH or PEM format for user accounts.

Use this procedure to 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

Before beginning this procedure, you must have done the following:

- Logged in to the CLI in EXEC mode
- Generated 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**	Places you into 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 you to EXEC mode.
Step 5	switch# **show user-account**	(Optional) Displays the user account configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to start SSH sessions using IP to connect to remote devices.

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Obtained the hostname and, if needed, the username, for the remote device.
- Enabled the SSH server on the remote device

Procedure

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

```
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

Use this procedure to 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.

```
switch# clear ssh hosts
```

Disabling the SSH Server

Use this procedure to disable the SSH server to prevent SSH access to the switch. By default, the SSH server is enabled.

If you disable SSH, to enable it again you must first generate an SSH server key

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**	Places you into global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server. The default is enabled.
Step 3	switch(config)# **show ssh server**	(Optional) Displays the SSH server configuration.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to delete SSH server keys after you disable the SSH server.

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

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**	Places you into 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.

	Command or Action	Purpose
		The default is to delete all the SSH keys.
Step 4	switch(config)# **show ssh key**	(Optional) Displays the SSH server key configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to clear SSH sessions from the device.

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

```
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 one of 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 the steps you use 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	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 7

Configuring Telnet

This chapter contains the following sections:

- Information About the Telnet Server , page 73
- Prerequisites for Telnet, page 73
- Guidelines and Limitations for Telnet, page 73
- Default Setting for Telnet, page 74
- Configuring Telnet, page 74
- Verifying the Telnet Configuration, page 75
- Feature History for Telnet, page 76

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 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.

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	Disabled

Configuring Telnet

Enabling the Telnet Server

The Telnet server is disabled by default, but you can use this procedure to enable it if necessary.

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**	Places you into global configuration mode.
Step 2	switch(config)# **feature telnet**	Enables the Telnet server.
Step 3	switch(config)# **show telnet server**	(Optional) Displays the Telnet server configuration.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Starting an IP Telnet Session to a Remote Device

Before You Begin

Before beginning this procedure, you must have done the following:

- Logged in to the CLI in EXEC mode
- Verified that the Telnet server is enabled and it is also enabled on the remote device

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

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet** {*ip 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*—The port number, from 1 to 65535, to use for this session. The default port number is 23 • *vrf-name*—The default VRF is the default VRF.

```
switch# telnet 10.10.1.1
```

Clearing Telnet Sessions

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

```
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 one of 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
Telnet	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 8

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

- Information About ACLs , page 77
- ACL Types and Applications , page 78
- Order of ACL Application, page 78
- Rules, page 78
- Statistics, page 80
- Prerequisites for IP ACLs, page 81
- Guidelines and Limitations for IP ACLs, page 81
- Default Settings for IP ACLs, page 81
- Configuring IP ACLs, page 81
- Verifying the IP ACL Configuration, page 89
- Monitoring IP ACLs, page 89
- Configuration Example for IP ACL, page 89
- Feature History for IP ACLs, page 90

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.

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 for Hyper-V 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 for Hyper-V 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 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 for Hyper-V 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.

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 ACL on the device and add rules to it.

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**	Places you into global configuration mode.
Step 2	switch(config)# [**no**] **ip access-list** *name*	Creates the named IP ACL (up to 64 characters in length) and enters IP ACL configuration mode. The no option removes the specified access list.
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.

	Command or Action	Purpose
		The permit and deny keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for Hyper-V Command Reference* for more information.
Step 4	switch(config-acl)# **statistics per-entry**	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	switch(config-acl)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 6	switch(config-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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
```

Changing an IP ACL

You can add and remove rules in an existing IPv4 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and 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

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**	Places you into global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Places you in IP ACL configuration mode for the specified ACL.
Step 3	switch(config-acl)# [*sequence-number*] { **permit** | **deny**} *protocol source destination*	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL.

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

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-acl)# 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, be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty, that is, an empty ACL with an implicit rule of "deny ip any any." Use the **show ip access-lists** command with the summary keyword to find the interfaces on which the IP ACL is configured.

Before You Begin

Before beginning this procedure, be sure that you have done the following:

- Logged 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**	Places you into global configuration mode.
Step 2	switch(config)# **no ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# **show ip access-list** *name* **summary**	(Optional) 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.

```
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

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**	Places you into 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.

	Command or Action	Purpose
Step 3	switch(config)# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
Enter configuration commands one command per line. End with CNTL/Z.
switch(config)# show ip access-list 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# copy running-config startup-config
```

Applying an IP ACL as a Port ACL

You can apply an IPv4 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 attached to a physical Ethernet interface or virtual Ethernet interface.

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

Before You Begin

Before beginning this procedure, be sure of the following:

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **interface {ethernet | vethernet}** *port*	Places you into interface configuration mode for the specified interface.

	Command or Action	Purpose
		Note: Port ACLs are not supported on a port-channel interface and physical Ethernet interface that is a member of the port-channel.
Step 3	switch(config-if)# **ip port access-group** *access-list* [**in** | **out**]	Applies an inbound or outbound IPv4 ACL to the interface. You can apply one port ACL to an interface.
Step 4	switch(config-if)# **show running-config aclmgr**	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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

!Command: show running-config aclmgr
!Time: Wed Mar 13 02:19:05 2013

version 5.2(1)SM1(5.1)
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# copy running-config startup-config

version 5.2(1)SM1(5.1)
```

Adding an IP ACL to a Port Profile

You can use this procedure to add an IP 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

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Created 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**	Places you into 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 access-group** *name* { **in** | **out** }	Adds the named ACL to the port profile for either inbound or outbound traffic.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# port-profile vm_eth1
switch(config-port-prof)# ip port access-group acl-01 out
switch(config-port-prof)# end
switch# 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# copy running-config startup-config
```

Applying an IP ACL to the Management Interface

Use this procedure to apply an IPv4 ACL to the Management interface, mgmt0.

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

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**	Places you into global configuration mode.
Step 2	switch(config)# **interface mgmt0**	Places you into interface configuration mode for the management interface.
Step 3	switch(config-if)# [**no**] **ip access-group***access-list* [**in** | **out**]	Applies a specified inbound or outbound IPv4 ACL to the interface. The no option removes the specified configuration.
Step 4	switch(config-if)# **show ip access-lists** *access-list*	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)#**[no] ip access-list match-local-traffic**	The match-local-traffic option enables matching for locally-generated traffic. Note: This global command must be enabled for ACL rules to take effect when an ACL is applied in egress direction on mgmt0 interface.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

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 access-lists [*name*]	Displays all IPv4 access control lists (ACLs) or a named IPv4 ACL.
show ip access-list [*name*] **summary**	Displays a summary of all configured IPv4 ACLs or a named IPv4 ACL.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.

Monitoring IP ACLs

Use one of the following commands for IP ACL monitoring:

Command	Purpose
show ip access-lists	Displays IPv4 ACL configuration. If the IPv4 ACL includes the statistics per-entry command, the show ip access-lists command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 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 a physical Ethernet interface that is not a member of a 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-acl# ip port access-group acl-01 in
switch(config-acl)# 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)#
```

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
IP ACLs	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 9

Configuring a MAC ACL

This chapter contains the following sections:

- Information About MAC ACLs, page 91
- Prerequisites for MAC ACLs, page 91
- Guidelines and Limitations for MAC ACLs, page 91
- Default Settings for MAC ACLs, page 92
- Configuring MAC ACLs, page 92
- Verifying MAC ACL Configurations, page 98
- Monitoring MAC ACLs, page 99
- Configuration Examples for MAC ACLs, page 99
- Feature History for MAC ACLs, page 100

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

Use this procedure to 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

Before beginning this procedure, you must be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Have a name to assign to the ACL you are creating.
- Created 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 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**	Places you into 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. See the *Cisco Nexus 1000V for Hyper-V Command Reference* for more information.

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

```
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# copy running-config startup-config
```

Changing a MAC ACL

Use this procedure to 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

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- In an existing MAC ACL, you cannot change existing rules.
- In an existing MAC ACL, you can add and remove rules.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*	(Optional) Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL.

	Command or Action	Purpose
		Without a sequence number, the rule is added to the end of the rules. The permit and deny keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for Hyper-V Command Reference* for more information.
Step 4	switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}	(Optional) Removes the rule that you specify from the MAC ACL. The permit and deny keywords support many ways of identifying traffic. See the *Cisco Nexus 1000V for Hyper-V Command Reference* for more information.
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	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) 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.

```
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. Be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are current 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.

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

Before beginning this procedure, be sure of the following:

- You are logged in to the CLI in EXEC mode.
- You know whether the ACL is applied to an interface.

Procedure

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

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

Changing Sequence Numbers in a MAC ACL

Use this procedure to 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

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**	Places you into 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

	Command or Action	Purpose
		preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	switch(config-mac-acl)# **show mac access-lists** *name*	(Optional) Displays the MAC ACL configuration for verification.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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 interface

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

Note: ACL cannot be applied on Port-channel interface. However it can be applied on a physical ethernet interface which is not part of the portchannel.

Before You Begin

Before beginning this procedure, be sure of the following:

- You are logged in to the CLI in EXEC mode
- 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**	Places you into 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	switch(config-if)# **show running-config aclmgr**	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface ethernet 1
switch(config-if)# mac port access-group acl-mac-01 in
switch(config-if)#  show running-config aclmgr

!Command: show running-config aclmgr
!Time: Wed Mar 13 03:38:02 2013

version 5.2(1)SM1(5.1)
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
version 5.2(1)SM1(5.1)
```

Adding a MAC ACL to a Port Profile

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Created 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**	Places you into 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	switch(config-port-prof)# **show port-profile name** *profile-name*	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
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.

Command	Purpose
show mac address-lists summary	Displays a summary of all configured MAC ACLs or a named MAC ACL.
show running-config aclmgr	Displays the ACL configuration, including MAC ACLs and the interfaces they are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL.

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 a MAC ACL named acl-mac-01 and apply it as a port ACL on a physical Ethernet interface that is not a member of a port-channel and configuration verification with match counters:

```
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)# 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	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 10

Configuring Port Security

This chapter contains the following sections:

- Information About Port Security, page 101
- Guidelines and Limitations for Port Security, page 105
- Default Settings for Port Security, page 106
- Configuring Port Security, page 106
- Verifying the Port Security Configuration, page 118
- Displaying Secure MAC Addresses, page 118
- Configuration Example for Port Security, page 118
- Feature History for Port Security, page 119

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.

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.

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:

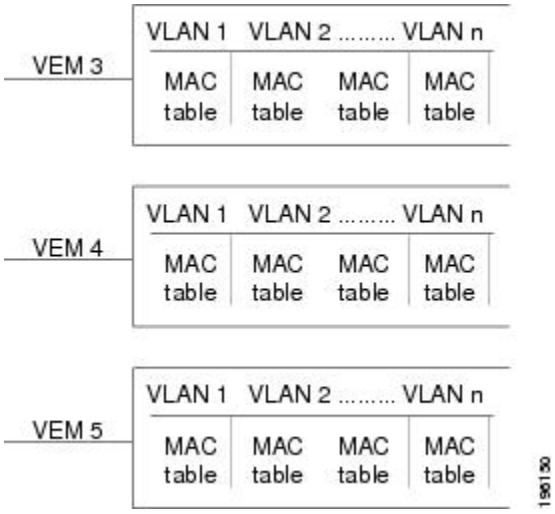
- Inactivity—The length of time after the device last received a packet from the address on the applicable interface.
- Absolute—The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

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 4: 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—The device has a nonconfigurable limit of 24,000 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.
- Interface maximum—You can configure a maximum number of secure MAC addresses for each interface protected by port security. The default interface maximum is one address 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.

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 either of the following occurs:

- 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.

- Ingress traffic from a secure MAC address arrives at a different interface in the same VLAN as the interface on which the address is secured.

Note After a secure MAC address is configured or learned on one secure port, the sequence of events that occurs when port security detects that secure MAC address on a different port in the same VLAN is known as a MAC move violation.

When a security violation occurs 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.

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.

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.

	Command or Action	Purpose
Step 4	switch(config-if)# **show port-security address interface vethernet number**	(Optional) Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	(Optional) Displays the port security configuration on the interface.
Step 6	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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**	Places you into 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**	(Optional) Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	(Optional) Displays the port security configuration on the interface.
Step 6	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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.

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Determined if the interface maximum has been reached for secure MAC addresses. You can use the **show port-security** command.
- Enabled port security on the interface that you are configuring.
- By default, no static secure MAC addresses are configured on an interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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**	(Optional) Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	(Optional) Displays the port security configuration on the interface.
Step 6	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

```
switch# configure terminal
switch(config)# interface vethernet 36
switch(config-if)# switchport port-security mac-address 0019.D2D0.00AE
switch(config)# show running-config port-security
interface Vethernet36
switchport port-security
switchport port-security maximum 5
switchport port-security mac-address 0019.D2D0.00AE
switch(config)# show port-security address interface vethernet 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
```

```
---- ----------- ------ ----- ---------------
2304 0019.D2D0.00AE STATIC Vethernet36 0
2304 0050.5687.3C4B DYNAMIC Vethernet36 0
----------------------------------------------------------------------
VLAN MAC Address Type Age Port Mod
switch(config-if)# copy running-config startup-config
```

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

Use this procedure to remove a static or a sticky secure MAC address from a Layer 2 interface.

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode
- Enabled port security on the interface that you are configuring

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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**	(Optional) Displays the secure MAC address learnt on the interface.
Step 5	switch(config-if)# **show port-security interface vethernet number**	(Optional) Displays the port security configuration on the interface.
Step 6	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

```
switch(config-if)# interface Vethernet36
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security maximum 5
switch(config-if)# show port-security address interface vethernet 36
Secure Mac Address Table
----------------------------------------------------------------------
```

```
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0050.5687.1111 STATIC Vethernet36 0
2303 0050.5687.3C4B DYNAMIC Vethernet36 0
----------------------------------------------------------------------
switch(config-if)# no switchport port-security mac-address 0050.5687.1111

switch(config-if)# show port-security address interface vethernet 36
Secure Mac Address Table
----------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0050.5687.3C4B DYNAMIC Vethernet36 0
----------------------------------------------------------------------
```

Removing a Dynamic Secure MAC Address

Use this procedure to remove a specific address learned by the dynamic method or to remove all addresses learned by the dynamic method on a specific interface.

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

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**	Places you into global configuration mode.
Step 2	switch(config)# **clear port-security dynamic {interface vethernet** *number* | **address** *address*} [**vlan** *vlan-ID*]	Removes dynamically learned, secure MAC addresses, as specified. The keywords and arguments 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. • **vlan—** removes an address or addresses on a particular VLAN.
Step 3	switch(config)# **show port-security address**	(Optional) Displays secure MAC addresses.

This example shows how to remove a dynamically learned, secure MAC address:

```
 switch(config)# show port-security address interface vethernet 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0000.1111.2224 STATIC Vethernet36 0
2303 0050.5687.3C4B DYNAMIC Vethernet36 0
------------------------------------------------------------------------
switch(config)# clear port-security dynamic interface vethernet 36
switch(config)# show port-security address interface vethernet 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0000.1111.2224 STATIC Vethernet36 0
------------------------------------------------------------------------
```

Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure is 4096 addresses.

The secure MAC addresses share the Layer 2 Forwarding Table (L2FT). The forwarding table for each VLAN can hold up to 1024 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.

	Command or Action	Purpose
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 highest valid number is 4096. The **no** option resets the maximum number of MAC addresses to the default, which is 1. If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.
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	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) 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.

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

```
switch(config-if)# interface Vethernet36
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security maximum 425
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 : 425
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Security violation count : 0
switch(config-if)# show running-config port-security
interface Vethernet36
  switchport port-security
  switchport port-security maximum 425
```

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

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode
- Enabled port security on the interface that you are configuring
- By default, the aging time is 0 minutes, which disables aging.
- Absolute aging is the default aging type.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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	switch(config-if)# **show port-security address interface vethernet number**	(Optional) Displays the secure MAC address learnt on the interface.
Step 6	switch(config-if)# **show port-security interface vethernet number**	(Optional) Displays the port security configuration on the interface.
Step 7	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 8	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

```
switch(config-if)# show running-config port-security
interface Vethernet36
  switchport port-security
  switchport port-security aging type inactivity
  switchport port-security aging time 120
switch(config-if)# interface Vethernet36
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 36
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2304 0050.5687.3C4B DYNAMIC Vethernet36 120
------------------------------------------------------------------------
switch(config-if)# show port-security interface vethernet 36
Port Security : Enabled
Port Status : Secure UP
Violation Mode : Shutdown
Aging Time : 120 mins
Aging Type : Inactivity
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Security violation count : 0
```

Configuring a Security Violation Action

Use this procedure to 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

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode
- Enabled port security on the interface that you are configuring
- The default security action is to shut down the port on which the security violation occurs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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 and arguments 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.
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	switch(config-if)# **show running-config port-security**	(Optional) Displays the port security configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a security violation action:

```
switch(config-if)# show running-config port-security
interface Vethernet36
  switchport port-security
  switchport port-security violation protect
switch(config-if)# interface Vethernet36
switch(config-if)# switchport port-security
switch(config-if)# switchport port-security violation protect
switch(config-if)# show port-security interface vethernet 36
Port Security : Enabled
```

```
Port Status : Secure UP
Violation Mode : Protect
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
```

Recovering Ports Disabled for Port Security Violations

Use this procedure to 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

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**	Places you into 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)# **errdisable recovery cause psecure-violation**	Enables a timed automatic recovery of the specified port that is disabled for a port security violation.
Step 4	switch(config-if)# **errdisable recovery interval** *seconds*	Configures a timer recovery interval in seconds from 30 to 65535 seconds.

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

```
switch# configure terminal
switch(config)# interface vethernet 36
switch(config-if)# errdisable recovery cause psecure-violation
switch(config-if)# errdisable recovery interval 30
switch(config-if)# copy running-config startup-config
switch(config-if)# show errdisable recovery
ErrDisable Reason Timer Status
----------------- ------------
link-flap disabled
dhcp-rate-limit disabled
arp-inspection disabled
security-violation disabled
psecure-violation enabled
failed-port-state enabled
ip-addr-conflict disabled

Timer interval: 30
```

Verifying the Port Security Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status.
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.

Use the **show port-security address interface vethernet id** command to display all secured MAC addresses on that interface.

Configuration Example for Port Security

This example shows a port security configuration for the vEthernet 36 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# config terminal
switch(config)# interface vethernet 36
switch(config-if)# switchport port-security
switch(config-if))#  switchport port-security maximum 10
switch(config-if))#  switchport port-security maximum 7 vlan 10
switch(config-if))#  switchport port-security maximum 3 vlan 20
switch(config-if))#  switchport port-security violation protect
switch(config-if))#  switchport mode trunk
  switch(config-if)# show running-config interface vethernet 36
switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation protect
switchport mode trunk
```

The following example shows a port securtiy configuration for the vEthernet 40 interface as an access port with an interface maximum set to 20, a violation set to restrict, an absolute timeout of 1 minute and a port security static MAC address of 0000.1111.5555:

```
switch# config terminal
switch(config)# interface vethernet 40
switch(config-if)# switchport port-security aging time 1
switch(config-if)# switchport port-security aging type absolute
switch(config-if)# switchport port-security
```

```
switch(config-if)# switchport port-security maximum 20
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 40
  switchport port-security aging time 1
  switchport port-security aging type absolute
  switchport port-security
  switchport port-security maximum 20
  switchport port-security mac-address 0000.1111.5555
  switchport port-security violation restrict
switch(config-if)# show port-security interface vethernet 40
Port Security : Enabled
Port Status : Secure UP
Violation Mode : Restrict
Aging Time : 1 mins
Aging Type : Absolute
Maximum MAC Addresses : 20
Total MAC Addresses : 2
Configured MAC Addresses : 1
Sticky MAC Addresses : 0
Security violation count : 0
```

This example shows a port security configuration for the vEthernet 42 interface as an access port with a violation set to shutdown and MAC address learning set to sticky:

```
switch# config terminal
switch(config)# interface vethernet 42
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)# show running-config interface vethernet 42
  switchport port-security
  switchport port-security mac-address sticky
  switchport port-security violation shutdown

switch(config-if)# show port-security interface vethernet 42
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 : 1
Security violation count : 0

switch(config-if)# show port-security address interface vethernet 42
Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address Type Ports Configured Age
(mins)
---- ----------- ------ ----- ---------------
2303 0050.5687.3C68 STICKY Vethernet42 0
------------------------------------------------------------------------
```

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
Port Security	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 11

Configuring DHCP Snooping

This chapter contains the following sections:

- Information About DHCP Snooping, page 121
- DHCP Overview, page 122
- BOOTP Packet Format, page 124
- Trusted and Untrusted Sources, page 126
- DHCP Snooping Binding Database, page 127
- DHCP Snooping Option 82 Data Insertion, page 127
- Licensing Requirements for DHCP Snooping, page 129
- Prerequisites for DHCP Snooping, page 130
- Guidelines and Limitations for DHCP Snooping, page 130
- Default Values for DHCP Settings, page 130
- Configuring DHCP Snooping, page 131
- Verifying the DHCP Snooping Configuration, page 143
- Monitoring DHCP Snooping , page 143
- Configuration Example for DHCP Snooping, page 143
- Configuration Example for Trust Configuration and DHCP Server Placement in the Network, page 145
- Standards, page 147
- Feature History for DHCP Snooping, page 147

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.

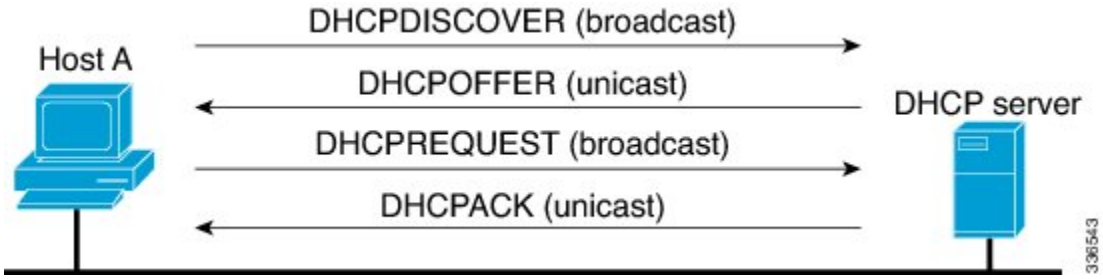
DHCP operations are categorized into four basic phases:

- 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 5: 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 6: BOOTP Packet Format

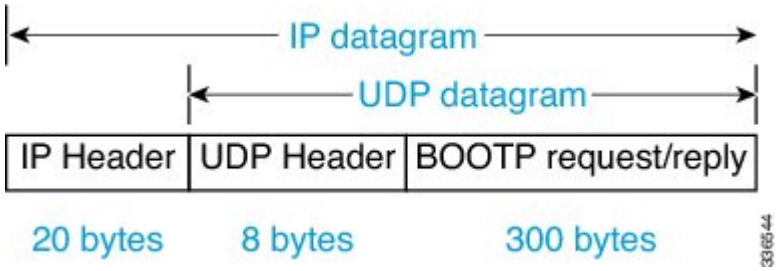


Figure 7: 300-Byte BOOTP Request and Reply Format

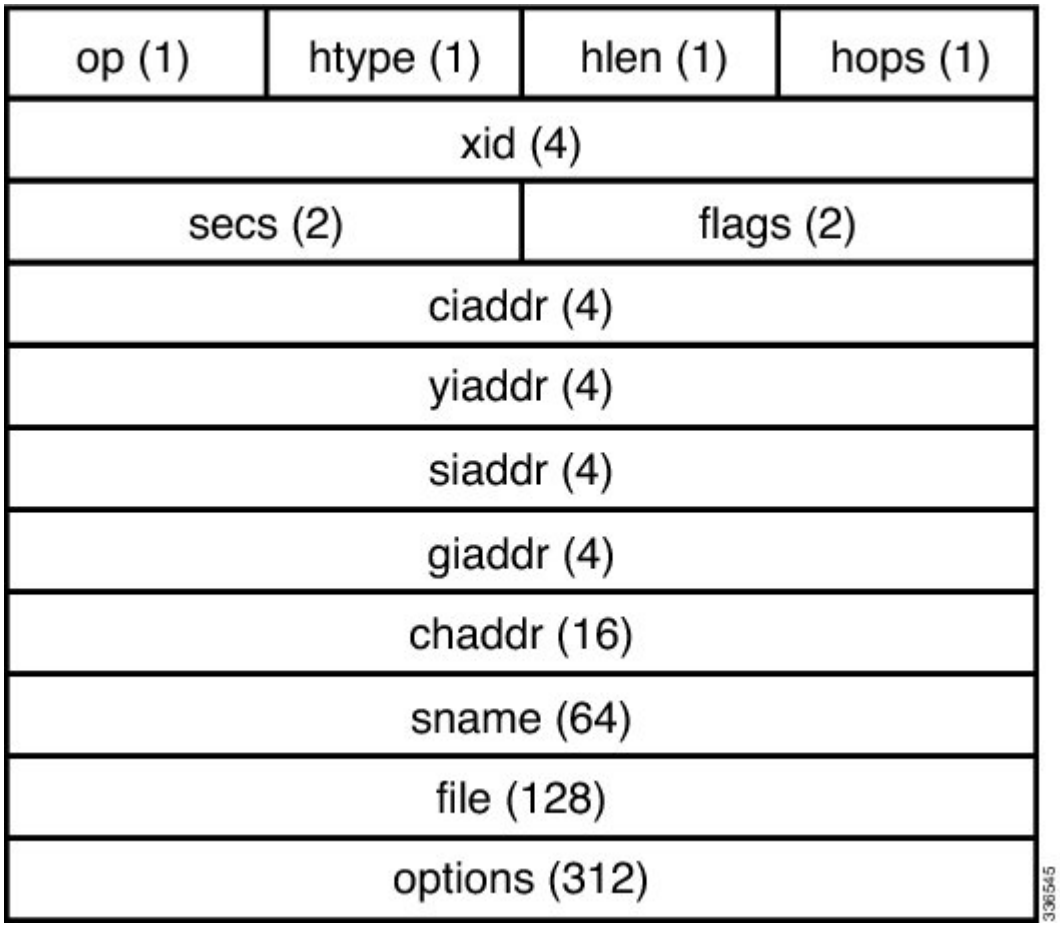


Table 3: BOOTP Request and Reply Format

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.
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.

Field	Bytes	Name	Description
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.
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.

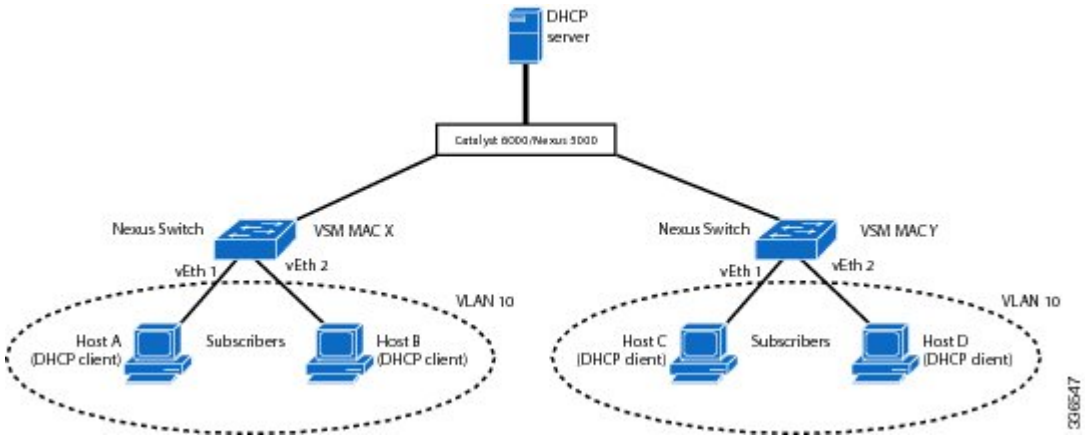
Option 82 Insertion

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 8: 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 9: Remote ID Suboption Frame Format

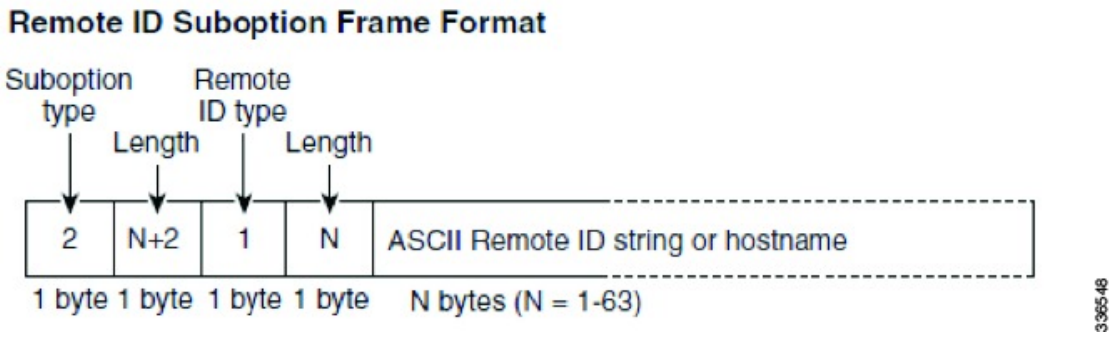
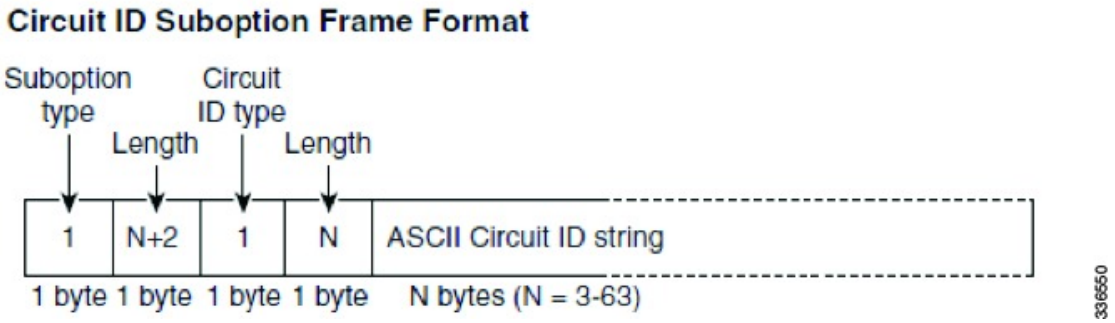


Figure 10: Circuit ID Suboption Frame Format



Licensing Requirements for DHCP Snooping

The following table shows the licensing requirements for this feature:

Feature	License Requirement
DHCP Snooping	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** Enable DHCP Snooping on the Cisco Nexus 1000V using **feature dhcp** command. If the switch edition is Essential, the **feature** command fails.

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

- A DHCP snooping database is stored on each VEM and can contain up to 2048 bindings. The combined number of DHCP bindings entries from all VEMs is a maximum of 12,000.
- 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.
- If you are configuring more than 128 access control lists (ACL) (MAC and IP ACLs combined), make sure that the VSM RAM is set at 3 GB (3072 MB).
- You cannot enable DHCP snooping on VXLAN ports.

Default Values 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

Minimum 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

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**	Places you into global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables the feature globally. The no command option followed by a copy running-start command deletes all DHCP-related configurations. If you use the no command but don't use the copy running-start command, DHCP-related configurations remain stored in the startup configuration. DHCP Snooping is available as an advanced feature that requires a license. See the Cisco Nexus 1000V License Configuration Guide for more information on the licensing requirements for Cisco Nexus 1000V.
Step 3	switch(config)# **show feature**	(Optional) Displays the state (enabled or disabled) of each available feature.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
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**	Places you into 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	switch(config)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 5	switch(config)# **show ip dhcp snooping**	(Optional) Displays the DHCP snooping IP configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# ip dhcp snooping
switch(config)# show running-config dhcp
feature dhcp ip dhcp snooping
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
Vethernet4            No              Unlimited
```

```
Vethernet5            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

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**	Places you into 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 vlan-list. The **no** option disables DHCP snooping on the VLANs specified.
Step 4	switch(config)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 5	switch(config)# **show ip dhcp snooping**	(Optional) Displays the DHCP snooping IP configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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
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
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
Vethernet4            No              Unlimited
Vethernet5            No              Unlimited


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

Enabling or Disabling DHCP Snooping MAC Address Verification

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

Before You Begin

You must be logged in to the CLI in EXEC mode.

Procedure

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

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

```
switch# configure terminal
switch(config)# no ip dhcp snooping verify mac-address
switch(config)# show running-config dhcp
feature dhcp
```

```
ip dhcp snooping
no ip dhcp snooping verify mac-address
ip dhcp snooping vlan 100,200,250-252
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 disabled
DHCP snooping trust is configured on the following interfaces:
Interface             Trusted        Pkt Limit
------------          -------        ---------
Vethernet1            No              Unlimited
Vethernet2            No              Unlimited
Vethernet3            No              Unlimited
Vethernet4            No              Unlimited
Vethernet5            No              Unlimited
switch(config)# copy running-config startup-config
```

Configuring an Interface as Trusted or Untrusted

Use this procedure to 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, vEthernet 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

- You are logged in to the CLI in EXEC mode.
- 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 the 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	switch(config-if)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# interface vethernet 3
switch(config-if)# ip dhcp snooping trust

switch(config)# port-profile vm-data
switch(config-port-profile)# ip dhcp snooping trust
switch(config-port-profile)# show running-config dhcp
feature dhcp
interface Vethernet1
  ip dhcp snooping trust
interface Vethernet3
  ip dhcp snooping trust
interface Vethernet10
  ip dhcp snooping trust
interface Vethernet11
  ip dhcp snooping trust
interface Vethernet12
  ip dhcp snooping trust
interface Vethernet13
  ip dhcp snooping trust
ip dhcp snooping
no ip dhcp snooping verify mac-address
ip dhcp snooping vlan 100,200,250-252
switch(config-port-profile)# copy running-config startup-config
```

Configuring the Rate Limit for DHCP Packets

Use this procedure to configure a limit for the rate of DHCP packets per second received on each port.

Before You Begin

Before beginning this procedure, you must be logged 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 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.
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in the 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	switch(config-if)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure 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
switch(config-if)# copy running-config startup-config

switch(config)# port-profile vm-data
switch(config-port-profile)# ip dhcp snooping limit rate 15
switch(config-port-profile)# show running-config dhcp
feature dhcp
interface Vethernet3
  ip dhcp snooping trust
  ip dhcp snooping limit rate 15
ip dhcp snooping
no ip dhcp snooping verify mac-address
ip dhcp snooping vlan 100,200,250-252
switch(config-port-profile)# 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
Vethernet4            No              Unlimited
Vethernet5            No              Unlimited
switch(config-port-profile)# copy running-config startup-config
```

Detecting Ports Disabled for DHCP Rate Limit Violation

Use this procedure to globally configure detection of ports disabled for exceeding 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

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**	Places you into 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	switch(config)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 5	switch(config)# **show errdisable detect**	(Optional) Displays the reasons for the port to be in the error-disabled state.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.

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 running-config dhcp
switch(config)# show errdisable detect
ErrDisable Reason               Timer Status
-----------------               ------------
link-flap                       enabled
dhcp-rate-limit                 enabled
arp-inspection                  enabled
ip-addr-conflict                enabled
switch(config)# copy running-config startup-config
```

Recovering Ports Disabled for DHCP Rate Limit Violations

Use this procedure to globally configure automatic recovery of ports disabled for violating the DHCP rate limit.

Ports that rate cause the port to be put into an errdisable state.

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

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**	Places you into 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 *timer interval* is the number of seconds from 30 to 65535.
Step 4	switch(config)# **show errdisable recovery**	(Optional) Displays the recovery interval for the vEth to recover from the error-disabled state.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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)# errdisable recovery interval 30
switch(config)# show running-config dhcp
switch(config)# show errdisable recovery
ErrDisable Reason               Timer Status
-----------------               ------------
link-flap                       disabled
dhcp-rate-limit                 enabled
arp-inspection                  disabled
security-violation              disabled
psecure-violation               disabled
failed-port-state               enabled
ip-addr-conflict                disabled

        Timer interval: 30
switch(config)# copy running-config startup-config
```

Clearing the DHCP Snooping Binding Database

Clearing All Binding Entries

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# **clear ip dhcp snooping binding**	Clears dynamically added entries from the DHCP snooping binding database.
Step 2	switch# **show ip dhcp snooping binding**	(Optional) Displays the DHCP snooping binding database.

```
switch# clear ip dhcp snooping binding
switch# show ip dhcp snooping binding
```

Clearing Binding Entries for an Interface

Before You Begin

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode
- Collected 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.

```
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 vEth port information in DHCP packets. This process is also called option 82 and the Relay Agent Information option.

Before You Begin

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

Note In a HA pair setup , the MAC inserted in the option 82 field of the DHCP packet is of the AIPC interface of the current ACTIVE VSM. Hence, the match criteria on the DHCP server will need to match the AIPC MAC of both Primary and Secondary VSM taking switchover into account.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into 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	switch(config)# **show running-config dhcp**	(Optional) Displays the DHCP snooping configuration.
Step 4	switch(config)# **show ip dhcp snooping**	(Optional) Displays the DHCP snooping IP configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to 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
no ip dhcp snooping verify mac-address
ip dhcp snooping vlan 100,200,250-252
```

```
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
Vethernet4            No              Unlimited
Vethernet5            No              Unlimited
switch(config)# copy running-config startup-config
```

Adding or Removing a Static IP Entry

By default, there are no static IP source entries on a device. Use this procedure to add or remove a static IP entry on a Cisco Nexus1000.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables IPSG feature. IPSG is available as an advanced feature that requires a license.
Step 3	switch(config)# [**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface vethernet** *interface-number*	Creates a static IP source entry for the current interface, or if you use the no option, removes a static IP source entry.
Step 4	switch(config)# **show ip dhcp snooping binding** [**interface vethernet** *interface-number*]	(Optional) Displays IP-MAC address bindings for the interface specified, including static IP source entries. Static entries appear with the term “static” in the Type column
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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 one of 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.

```
swtich(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 15pps on the interface where the client is connected, with clients using port-profile client-pp, and 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 veth 5
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)# errdisable detect cause dhcp-rate-limit
switch(config)# errdisable recovery interval 60
switch(config)# ip source binding 192.168.0.55  00:50:56:81:42:74 vlan 100 interface vethernet
```

```
 12

switch (config-if)# show feature
Feature Name          Instance      State
--------------------        -------      --------
cts                       1         disabled
dhcp-snooping             1         enabled
http-server               1         enabled
lacp                      1         enabled
netflow                   1         enabled
network-segmentation      1         enabled
port-profile-roles        1         disabled
private-vlan              1         enabled
segmentation              1         enabled
sshServer                 1         enabled
tacacs                    1         disabled
telnetServer              1         disabled
vtracker                  1         disabled

switch(config-if)# show run dhcp

feature dhcp

interface Vethernet1
  ip dhcp snooping limit rate 15

interface Vethernet5
  ip dhcp snooping trust

interface Vethernet10
  ip dhcp snooping limit rate 15

interface Vethernet11
  ip dhcp snooping limit rate 15

interface Vethernet12
  ip dhcp snooping limit rate 15

interface Vethernet13
  ip dhcp snooping limit rate 15
ip dhcp snooping
ip dhcp snooping vlan 100
ip source binding 192.168.0.55  00:50:56:81:42:74 vlan 100 interface vethernet 12

Note: Client interfaces Vethernet 1,10-13 are part of port-profile “client-pp”

swtich (config-if)# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on the following VLANs:
100
DHCP snooping is operational on the following VLANs:
100
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                     15
Vethernet2            No                     Unlimited
Vethernet3            No                     Unlimited
Vethernet4            No                     Unlimited
Vethernet5            Yes                    Unlimited
Vethernet7            No                     Unlimited
Vethernet8            No                     Unlimited
Vethernet9            No                     Unlimited
Vethernet10           No                    15
Vethernet11           No                    15
Vethernet12           No                    15
Vethernet13           No                    15


switch# show ip dhcp snooping binding
MacAddress              IpAddress        LeaseSec  Type        VLAN  Interface
```

```
------------------      ---------------  ---------- ----------- ----- -------------
00:50:56:81:42:46       192.168.0.9      28570     dhcp-snoop  100    Vethernet1
00:50:56:81:42:59       192.168.0.69     28591     dhcp-snoop  100    Vethernet10
00:50:56:81:42:6d       192.168.0.251    28559     dhcp-snoop  100    Vethernet11
00:50:56:81:42:72       192.168.0.48     infinite  static      100    Vethernet12
00:50:56:81:42:74       192.168.0.55     infinite  dhcp-snoop  100    Vethernet13
```

Note An entry with infinite lease time issued by the DHCP server will have infinite in the Lease Sec column and will be of Type dhcp-snoop.

When client interfaces are part of secondary VLAN, the DHCP binding table displays the entries on its corresponding primary VLAN.

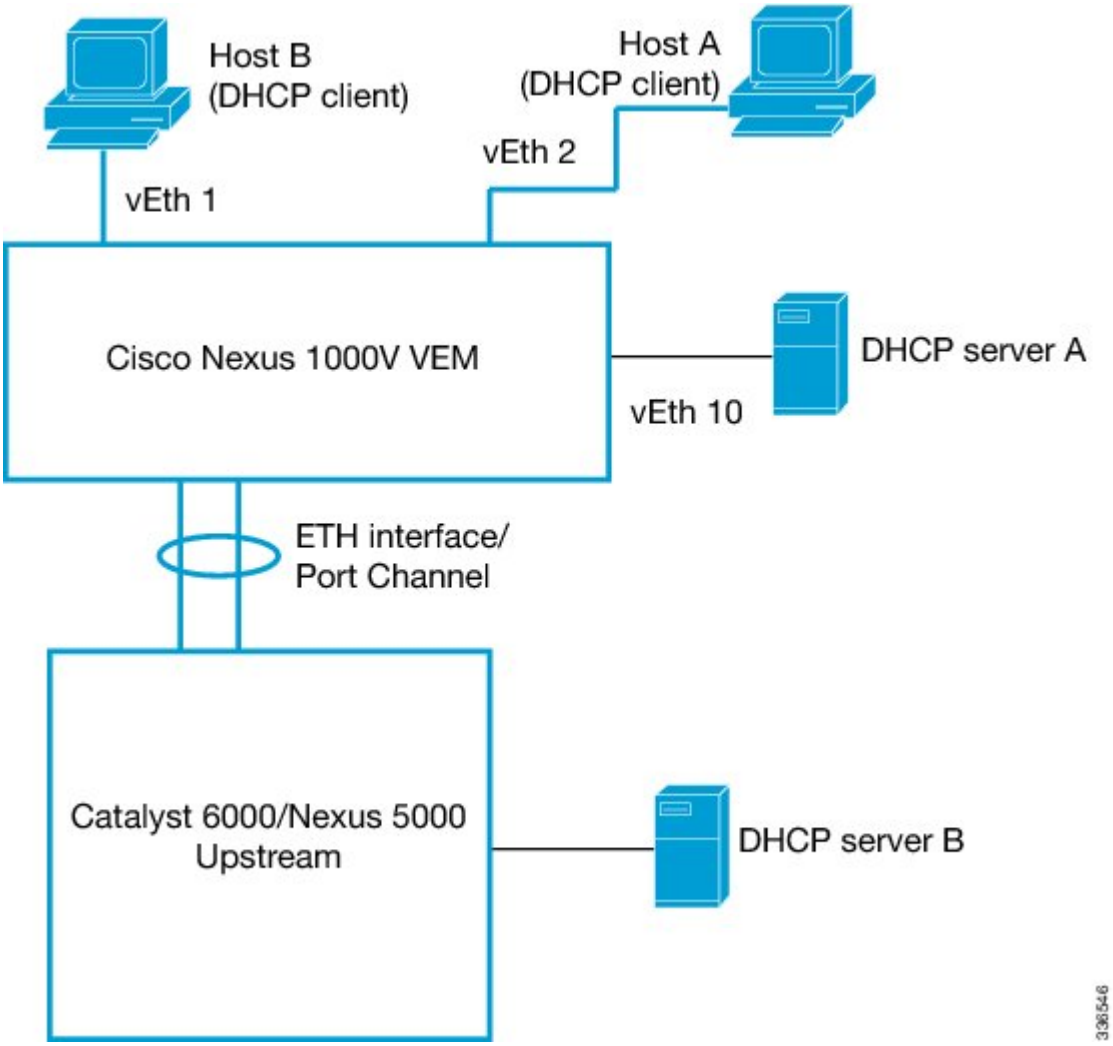
Configuration Example for Trust Configuration and DHCP Server Placement in the Network

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 11: DHCP Server Inside and Outside the Cisco Nexus 1000V Network and Clients on the Cisco Nexus 1000V

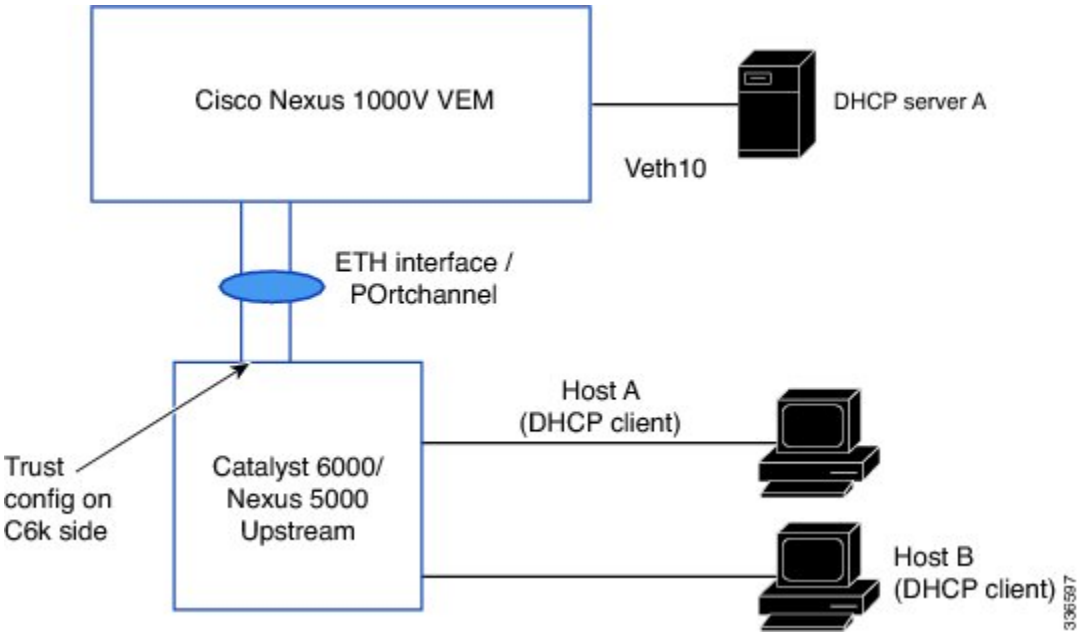


DHCP Server Inside the Cisco Nexus 1000V Network and Clients Outside the Cisco Nexus 1000V

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 12: 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
DHCP Snooping	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 12

Configuring Dynamic ARP Inspection

This chapter contains the following sections:

- Information About Dynamic ARP Inspection, page 149
- Prerequisites for DAI, page 152
- Guidelines and Limitations for DAI, page 152
- Default Settings for DAI, page 152
- Configuring DAI Functionality, page 153
- Verifying the DAI Configuration, page 162
- Monitoring DAI , page 162
- Configuration Examples for DAI, page 163
- Standards, page 166
- Feature History for DAI, page 166

Information About Dynamic ARP Inspection

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.

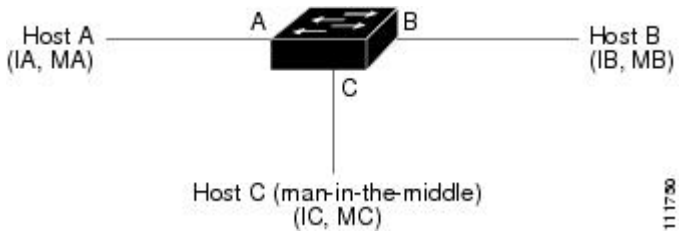
To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

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 13: 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.

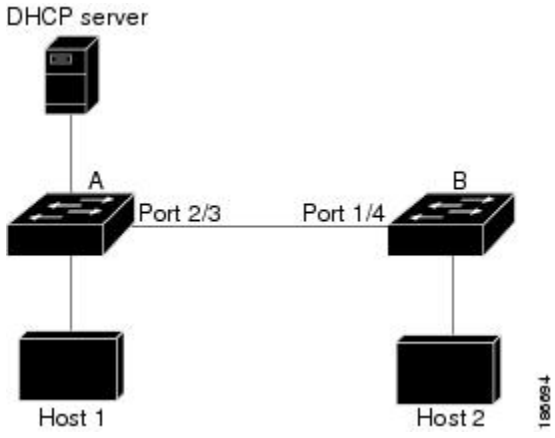
  Packets bypass all DAI validation checks.

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 14: 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	Untrusted.

Parameters	Default
Trust state of vEthernet interfaces	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	Unlimited.
Rate limit burst interval	1 second.
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**	Places you into global configuration mode.
Step 2	switch(config)# **ip arp inspection vlan** *list*	Configures the specified VLAN or list of VLANs for DAI.
Step 3	switch(config)# **show ip arp inspection vlan** *list*	(Optional) Displays the DAI status for the specified list of VLANs.

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

This example shows how to configure a VLAN for DAI:

```
switch# configure terminal
switch(config)# ip arp inspection vlan 13
switch(config)# show ip arp inspection vlan 13
Source Mac Validation      : Disabled
Destination Mac Validation      : Disabled
IP Address Validation      : Disabled

Filter Mode (For Static bindings): IP-MAC

Vlan : 13
-----------
Configuration        : Enabled
Operation State      : Active
DHCP logging options : Deny
switch(config)# copy running-config startup-config
```

Configuring a Trusted vEthernet Interface

Before You Begin

Before beginning this procedure, you must know the following information:

- By default, vEthernet interfaces are untrusted.
- 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 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

    If configuring a trusted interface on the port profile, it has already been created and you know its name.

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**	Places you into global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface vethernet** *interface-number*	Places you in the 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**	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.
Step 5	switch(config-if)# **ip arp inspection trust**	Configures the interface as a trusted ARP interface.
Step 6	switch(config-port-prof)# **ip arp inspection trust**	Configures the interfaces assigned to the port profile as trusted ARP interfaces.
Step 7	switch(config-if)# **show ip arp inspection interface vethernet** *interface-number*	(Optional) Displays the trusted state and the ARP packet rate for the specified interface.
Step 8	switch(config-if)# **show port-profile** *profilename*	(Optional) Displays the port profile configuration including the ARP trusted state.
Step 9	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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 interface vethernet 3
 Interface        Trust State        Pkt Limit      Burst Interval
 -------------    -----------        ---------      ----------------
 Vethernet3          Trusted             15                5
switch(config-if)# copy running-config startup-config

switch(config)# port-profile vm-data
switch(config-port-profile)#  ip arp inspection trust
switch(config-port-profile)#  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 13
  ip arp inspection trust
  no shutdown
 evaluated config attributes:
  switchport mode access
  switchport access vlan 13
  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-profile)#  copy running-config startup-config
```

Resetting a vEthernet Interface to Untrusted

By default, vEthernet interfaces are untrusted. Use this procedure to remove a trusted designation from a vEthernet interface, returning 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**	Places you into 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	switch(config-if)# **show ip arp inspection interface vethernet** *interface-number*	(Optional) Displays the trusted state and the ARP packet rate for the specified interface.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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—15 packets per second
- 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**	Places you into 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)# **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 and arguments are as follows: • rate—specifies that allowable values are between 1 and 2048 packets per second (pps). ◦ Untrusted interface default = 15 packets per second. ◦ Trusted interface default = 15 packets per second. • burst interval—specifies that allowable values are between 1 and 15 seconds (default = 1 second).

	Command or Action	Purpose
		• none—specifies an unlimited number of packets per second.
Step 5	switch(config-if)# **show ip arp inspection interface vethernet interface-number**	(Optional) Displays the trusted state and the ARP packet rate for the specified interface.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Use this procedure to set the rate limit of ARP requests and responses.

Before beginning this procedure, you must know the following:

- Because of their aggregation, trunk ports should be configured with higher rate limit.
- 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 = 15 packets per second
  - 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

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**	Places you into global configuration mode.
Step 2	switch(config)# **feature dhcp**	Enables DAI feature. DAI is available as an advanced feature that requires a license.
Step 3	switch(config)# **interface vethernet** *interface-number*	Places you in the interface configuration mode for the specified vEthernet interface.
Step 4	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 and arguments are as follows: • rate—Untrusted interface default = 15 packets per second. ◦ Untrusted interface default = 15 packets per second. ◦ Trusted interface default = unlimited. • burst interval—default = 5 second. • none—an unlimited number of packets per second.
Step 5	switch(config)# **show ip arp inspection interface vethernet interface-number**	(Optional) Displays the default ARP packet rate for the specified interface.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

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**	Places you into 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 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. *timer-interval*—allowable values are from 30 to 65535 seconds.
Step 5	switch(config)# **show errdisable detect**	(Optional) Displays the errdisable configuration.
Step 6	switch(config)# **show errdisable recovery**	(Optional) Displays the errdisable configuration.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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
dhcp-rate-limit                 enabled
arp-inspection                  enabled
ip-addr-conflict                enabled
11:22 AM
switch(config)# sh errdisable recovery
ErrDisable Reason               Timer Status
-----------------               ------------
link-flap                       disabled
dhcp-rate-limit                 disabled
arp-inspection                  enabled
security-violation              disabled
```

```
psecure-violation               disabled
failed-port-state               enabled
ip-addr-conflict                disabled

        Timer interval: 30
switch(config-if)# 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, and drops packets with an invalid MAC address.

- 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, and drops packets with invalid MAC addresses.

Note Whenever you configure a validation, any previous validation configuration is overwritten.

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**	Places you into global configuration mode.
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.

	Command or Action	Purpose
Step 3	switch(config)# **show ip arp inspection**	(Optional) Displays the DAI configuration.
Step 4	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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
```

Verifying the DAI Configuration

Use one of 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.
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#
```

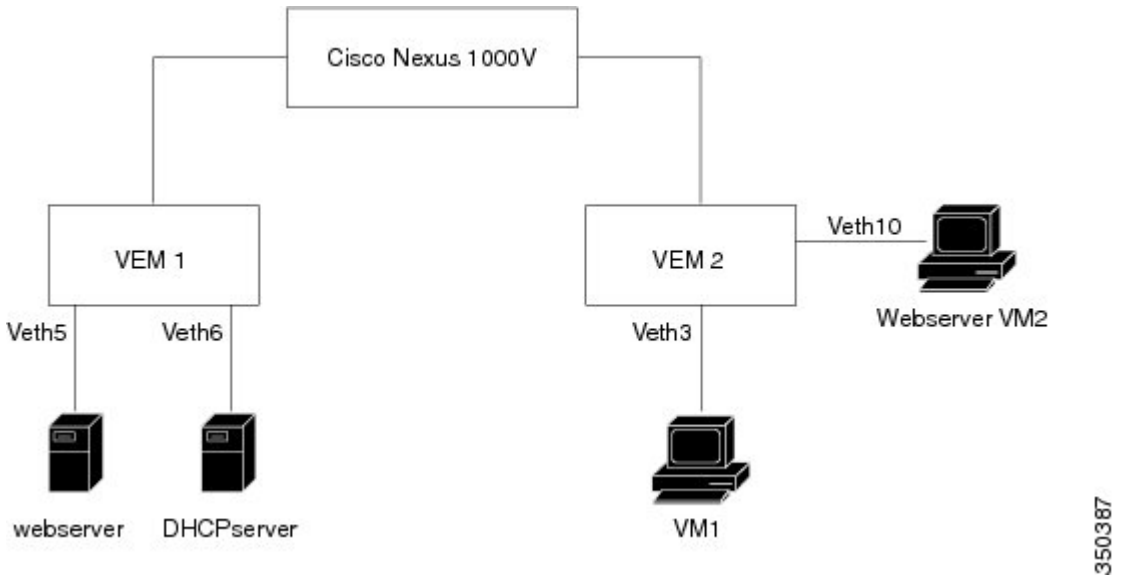
Configuration Examples for DAI

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 15: 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             15                5


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
```

Dropping an ARP Request Packet and Logging the Error Message

If VM 2 tries to send an ARP request with an IP address of 10.0.0.3, the packet is dropped and an error message is logged.

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on vEthernet3, vlan
1.([0002.0002.0002/10.0.0.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Fri Jul 13 2008])
```

Example of Displaying 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
DAI	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 13

Configuring IP Source Guard

This chapter includes the following sections:

- Information About IP Source Guard, page 167
- Prerequisites for IP Source Guard, page 168
- Guidelines and Limitations for IP Source Guard, page 168
- Default Settings for IP Source Guard, page 168
- Configuring IP Source Guard Functionality, page 169
- Verifying the IP Source Guard Configuration, page 170
- Monitoring IP Source Guard Bindings, page 170
- Configuration Example for IP Source Guard, page 170
- Feature History for IP Source Guard, page 170

Information About IP Source Guard

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

- Entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table.
- Static IP source entries that you configure.

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 the IP traffic when DHCP snooping adds a binding table entry for the IP address and MAC address of an IP packet or when you have configured a static IP source entry 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.

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.

Default Settings for IP Source Guard

Parameters	Default
IP Source Guard	Disabled on each interface.
IP source entries	None. No static or default IP source entries exist by default.

Configuring IP Source Guard 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**	Places you into 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	switch(config-if)# **show ip verify source interface vethernet interface number**	(Optional) Displays the IP Source Guard configuration.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

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
```

Verifying the IP Source Guard Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show running-config dhcp	Displays DHCP snooping configuration, including the IP Source Guard configuration.
show ip verify source	Displays IP-MAC address bindings.

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

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
```

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
IP Source Guard	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 14

Disabling HTTP Server

This chapter contains the following sections:

- Information About the HTTP Server, page 171
- Guidelines and Limitations for the HTTP Server, page 171
- Default Settings for the HTTP Server, page 171
- Disabling the HTTP Server, page 172
- Verifying the HTTP Configuration, page 172
- Related Documents for the Disabling the HTTP Server, page 173
- Standards, page 173
- Feature History for Disabling the HTTP Server, page 173

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 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

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**	Places you into global configuration mode.
Step 2	switch(config)# **no feature http-server**	Disables the HTTP server.
Step 3	switch(config)# **show http-server**	(Optional) Displays the HTTP server configuration (enabled or disabled).
Step 4	switch(config) **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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%
```

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	5.2(1)SM1(5.1)	This feature was introduced.

CHAPTER 15

Blocking Unknown Unicast Flooding

This chapter contains the following sections:

- Information About UUFB , page 175
- Guidelines and Limitations for UUFB, page 175
- Default Settings for UUFB, page 176
- Configuring UUFB, page 176
- Standards, page 178
- Configuration Example for Blocking Unknown Unicast Packets, page 179
- Feature History for UUFB, page 179

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 the ports of an application or VM by using MAC addresses other than the one given by Microsoft.
- 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

Use this procedure to globally block unknown unicast packets from flooding the forwarding path for the switch.

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**	Places you in global configuration mode.
Step 2	switch(config)# [**no**] **uufb enable**	Configures UUFB globally for the VSM.
Step 3	switch(config)# **show uufb status**	(Optional) Displays the UUFB global setting for the VSM.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to allow unknown unicast packets to flood a vEthernet interface if you have blocked flooding globally for the VSM. You can also use this procedure to 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

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**	Places you in global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters 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	switch(config)# **show running-config vethernet** *interface-number*	(Optional) Displays the running configuration for the interface for verification.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
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

Use this procedure to 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 use this procedure to 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

Before beginning this procedure, be sure you have done the following:

- Logged in to the CLI in EXEC mode.
- Configured the vEthernet port profile for which you want to allow flooding.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you in 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	switch(config-port-prof)# **show running-config port-profile** *profile-name*	(Optional) Displays the configuration for the named port profile for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# port-profile accessprof
switch(config-port-prof)# switchport uufb disable
```

Standards

Standards	Title
RFC-2131	Dynamic Host Configuration Protocol (http://tools.ietf.org/html/rfc2131)

Standards	Title
RFC-3046	DHCP Relay Agent Information Option (http://tools.ietf.org/html/rfc3046)

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	5.2(1)SM1(5.1)	This feature was introduced.