Send document comments to nexus1k-docfeedback@cisco.com.

Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1) SV1(4)

February 17, 2012

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

Text Part Number: OL-22823-01

Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1) SV1(4)

CONTENTS

New and Changed Information

This chapter lists the information that is new or was changed in this document per release, and where it is located.

Feature	Description	Changed in release	Where Documented
DHCP Snooping Relay Agent (Option 82)	You can configure DHCP to relay VSM MAC and port information in DHCP packets.	4.2(1)SV1(4)	Chapter 12, “Configuring DHCP Snooping”
DHCP Snooping binding table	You can clear DHCP snooping binding table entries for an interface.	4.2(1)SV1(4)	Chapter 12, “Configuring DHCP Snooping”
Enable DHCP	You can enable or disable DHCP globally using the **feature DHCP** command.	4.2(1)SV1(4)	Chapter 12, “Configuring DHCP Snooping”
Enable SSH server	You can enable or disable the SSH server using the **feature DHCP** command.	4.2(1)SV1(4)	Chapter 7, “Configuring SSH”
Enable Telnet server	You can enable or disable the Telnet server using the **feature DHCP** command.	4.2(1)SV1(4)	Chapter 8, “Configuring Telnet”
Disable HTTP Server	Disabling the HTTP server for security purposes.	4.2(1)SV1(4)	Chapter 15, “Disabling HTTP Server”
VSD	Virtual service domains (VSDs) allow you to classify and separate traffic for network services.	4.0(4)SV1(2)	Chapter 3, “Configuring VSD”
DHCP Snooping	Dynamic Host Configuration Protocol (DHCP) snooping acts like a firewall between untrusted hosts and trusted DHCP servers.	4.0(4)SV1(2)	Chapter 12, “Configuring DHCP Snooping”
Dynamic ARP Inspection (DAI)	Dynamic ARP inspection (DAI) provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address.	4.0(4)SV1(2)	Chapter 13, “Configuring Dynamic ARP Inspection”
IP Source Guard	IP Source Guard is a per-interface traffic permit filter for IP and MAC addresses.	4.0(4)SV1(2)	Chapter 14, “Configuring IP Source Guard”

Preface

The Security Configuration document provides procedures for configuring security features, such as AAA, VSD, SSH, and so forth.

This preface describes the following aspects of this document:

Audience

This guide is for experienced network system users.

Document Organization

This document is organized into the following chapters:

Chapter and Title	Description
Chapter 1, “Security Overview”	Describes the security features.
Chapter 2, “Managing User Accounts”	Describes how to configure user accounts.
Chapter 3, “Configuring VSD”	Describes how to configure VSD.
Chapter 4, “Configuring AAA”	Describes how to configure AAA.
Chapter 5, “Configuring RADIUS”	Describes how to configure RADIUS.
Chapter 6, “Configuring TACACS+”	Describes how to configure TACACS+.
Chapter 7, “Configuring SSH”	Describes how to configure SSH.
Chapter 8, “Configuring Telnet”	Describes how to configure Telnet.
Chapter 9, “Configuring an IP ACL”	Describes how to configure IP access control lists (ACLs) for filtering traffic.

Chapter and Title	Description
Chapter 10, "Configuring a MAC ACL"	Describes how to configure MAC access control lists (ACLs) for filtering traffic.
Chapter 11, "Configuring Port Security"	Describes how to configure port security.
Chapter 12, "Configuring DHCP Snooping"	Describes how to configure DHCP snooping.
Chapter 13, "Configuring Dynamic ARP Inspection"	Describes how to configure Dynamic ARP Inspection.
Chapter 14, "Configuring IP Source Guard"	Describes how to configure IP Source Guard.
Chapter 15, "Disabling HTTP Server"	Describes how to disable HTTP server.
Chapter 16, "Security Configuration Limits"	Describes configuration limits for security features.

Document Conventions

Command descriptions use these conventions:

boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
{ }	Elements in braces are required choices.
[ ]	Elements in square brackets are optional.
x | y | z	Alternative, mutually exclusive elements are separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

`screen font`	Terminal sessions and information the device displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
`italic screen font`	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions for notes and cautions:

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

Send document comments to nexus1k-docfeedback@cisco.com.

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

Available Documents

This section lists the documents used with the Cisco Nexus 1000 and available on Cisco.com at the following url:

http://www.cisco.com/en/US/products/ps9902/tsd_products_support_series_home.html

General Information

Cisco Nexus 1000V Documentation Roadmap, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Release Notes, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Compatibility Information, Release 4.2(1)SV1(4)

Cisco Nexus 1010 Management Software Release Notes, Release 4.2(1)SP1(2)

Install and Upgrade

Cisco Nexus 1000V Virtual Supervisor Module Software Installation Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Software Upgrade Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V VEM Software Installation and Upgrade Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1010 Virtual Services Appliance Hardware Installation Guide

Cisco Nexus 1010 Software Installation and Upgrade Guide, Release 4.2(1)SP1(2)

Configuration Guides

Cisco Nexus 1000V License Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V High Availability and Redundancy Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Quality of Service Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1010 Software Configuration Guide, Release 4.2(1)SP1(2)

Programming Guide

Cisco Nexus 1000V XML API User Guide, Release 4.2(1)SV1(4)

Send document comments to nexus1k-docfeedback@cisco.com.

Reference Guides

Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)

Cisco Nexus 1000V MIB Quick Reference

Cisco Nexus 1010 Command Reference, Release 4.2(1)SP1(2)

Troubleshooting and Alerts

Cisco Nexus 1000V Troubleshooting Guide, Release 4.2(1)SV1(4)

Cisco Nexus 1000V Password Recovery Guide

Cisco NX-OS System Messages Reference

Virtual Security Gateway Documentation

Cisco Virtual Security Gateway for Nexus 1000V Series Switch Release Notes, Release 4.2(1)VSG(1)

Cisco Virtual Security Gateway, Release 4.2(1)VSG1(1) and Cisco Virtual Network Management Center, Release 1.0.1 Installation Guide

Cisco Virtual Security Gateway for Nexus 1000V Series Switch License Configuration Guide, Release 4.2(1)VSG1(1)

Cisco Virtual Security Gateway for Nexus 1000V Series Switch Configuration Guide, Release 4.2(1)VSG1(1)

Cisco Virtual Security Gateway for Nexus 1000V Series Switch Command Reference, Release 4.2(1)VSG1(1)

Virtual Network Management Center

Release Notes for Cisco Virtual Network Management Center, Release 1.0.1

Cisco Virtual Security Gateway, Release 4.2(1)VSG1(1) and Cisco Virtual Network Management Center, Release 1.0.1 Installation Guide

Cisco Virtual Network Management Center CLI Configuration Guide, Release 1.0.1

Cisco Virtual Network Management Center GUI Configuration Guide, Release 1.0.1

Cisco Virtual Network Management Center XML API Reference Guide, Release 1.0.1

Network Analysis Module Documentation

Cisco Network Analysis Module Software Documentation Guide, 4.2

Cisco Nexus 1000V NAM Virtual Service Blade Installation and Configuration Guide

Network Analysis Module Command Reference Guide, 4.2

User Guide for the Cisco Network Analysis Module Virtual Service Blades, 4.2

Cisco Network Analysis Module Software Release Notes, 4.2

Obtaining Documentation and Submitting a Service Request

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

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

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

Send document comments to nexus1k-docfeedback@cisco.com.

CHAPTER 1

Security Overview

This chapter provides an overview of the following security features used with the Cisco Nexus 1000V:

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. For information about configuring and managing user accounts, see Chapter 2, “Managing User Accounts.”

Virtual Service Domain

A virtual service domain (VSD) allows you to classify and separate traffic for network services, such as firewalls, traffic monitoring, and those in support of compliance goals such as Sarbanes Oxley. For information about configuring and managing VSD, see Chapter 3, “Configuring VSD.”

Authentication, Authorization, and Accounting (AAA)

AAA, called Triple A, 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.

For information about configuring AAA, see Chapter 4, "Configuring AAA."

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.

For information about configuring RADIUS, see Chapter 5, "Configuring RADIUS."

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.

For information about configuring TACACS+, see Chapter 6, "Configuring TACACS+."

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.

For information, see the Chapter 7, "Configuring SSH."

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. For information, see the Chapter 8, "Configuring Telnet.".

Access Control Lists (ACLs)

An ACL is an ordered set of rules for filtering traffic. When the device determines that an ACL applies to a packet, it tests the packet against the rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies a default rule. The device processes packets that are permitted and drops packets that are denied.

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

For more information, see the following:

- Chapter 9, "Configuring an IP ACL"
- Chapter 10, "Configuring a MAC ACL"

Port Security

Port security lets you 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.

For more information, see Chapter 11, "Configuring Port Security."

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.

For more information, see Chapter 12, "Configuring DHCP Snooping."

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.

For more information, see Chapter 13, "Configuring Dynamic ARP Inspection."

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

For more information, see Chapter 14, "Configuring IP Source Guard."

CHAPTER 2

Managing User Accounts

This chapter describes how to configure user accounts and includes the following topics:

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

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**

Table 2-1 describes the components that make up a role.

Table 2-1 Role Components

Component	Description

Rule	One of the defined role criteria, such as a command that is permitted or denied. You can add up to 256 rules to each role.
	The following are the rules for the predefined roles:
	• **role: network-admin**
	`Rule Perm Type Scope Entity`
	`1 permit read-write`
	• **role: network-operator**
	`Rule Perm Type Scope Entity`
	`1 permit read-only`

Feature	An individual feature, such as syslog or TACACS+, whose access can be defined in a rule. To see a list of available features, use the **show role feature** command.
Feature Group	A grouping of features whose access can be defined in a rule. You can create up to 64 such groupings. To see a list of available feature groups, use the **show role feature-group** command.
Command	A single command, or group of commands collected in a regular expression, whose access can be defined in a rule. A role permitting access to a command takes precedence over a role that denies access to the command. For example, if a user is assigned a role that denies access to the configuration command, but is also assigned a role that permits access to this command, then access is permitted.

User Name

A user name identifies an individual user by a unique character string, such as daveGreen. User names are case sensitive and can consist of up to 28 alphanumeric characters. A user name consisting of all numerals is not allowed. If an all numeric user name 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 (>)

Table 2-2 lists the characteristics of strong passwords.

Table 2-2 Characteristics of strong passwords

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

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Check of Password Strength

The device checks password strength automatically by default. When you add a user name and password, the strength of the password is evaluated. If it is a weak password, then the error message below displays to notify you.

```
n1000v# config t
n1000v(config)# username daveGreen password davey
password is weak
Password should contain characters from at least three of the classes:
```

```
lower case letters,upper case letters, digits, and special characters
```

Password strength-checking can be disabled.

Expiration Date

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

Guidelines and Limitations

User access has the following configuration guidelines and limitations:

- You can create up to 64 roles in addition to the two predefined user roles.
- You can create up to 256 rules in a user role.
- You can create up to 64 feature groups.
- You can add up to 256 users.
- You can assign a maximum of 64 user roles to a user account.
- If you have a user account that has the same name as a remote user account on an AAA server, the user roles for the local user account are applied to the remote user, not the user roles configured on the AAA server.

Default Settings

Table 2-3 lists the default settings for user access.

Table 2-3 User Access Defaults

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

This section includes the following topics:

- Creating a Feature Group, page 2-10
- Configuring Interface Access, page 2-12
- Configuring VLAN Access, page 2-13

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.
- Checking password strength is enabled by default. This procedure can be used to enable it again should it become disabled.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**password strength-check** **Example:** `n1000v(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	**show password strength-check**
	
	Example:
	`n1000v# show password strength-check`
	`Password strength check enabled`
	`n1000v(config)#`
		(Optional) Displays the configuration for checking password strength.

Step 4	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Disabling the Check of Password Strength

Use this procedure to disable the check of password strength.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.
- Checking password strength is enabled by default. This procedure can be used to disable it.

SUMMARY STEPS

1. **config t**
2. **no password strength-check**
3. **show password strength-check**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**no password strength-check**
	
	Example:
	`n1000v(config)# no password strength-check`
	`n1000v(config)#`
		Disables password-strength checking.
		
		The default is enabled.

Step 3	**show password strength-check**
	
	Example:
	`n1000v# show password strength-check`
	`Password strength check not enabled`
	`n1000v(config)#`
		(Optional) Displays the configuration for checking password strength.

Step 4	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Creating a User Account

Use this procedure to create and configure a user account, defining access to the Cisco Nexus 1000V.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.

- 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	mtsuser	rpcuser
bin	gopher	news	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. For more information about roles, see the "Role" section on page 2-1.
- 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 the "Configuring a User Account with a Public Key" section on page 7-5.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**show role** **Example:** `n1000v(config)# show role`	(Optional) Displays the available roles that can be assigned to users. You can create a new user role with the "Creating a Role" procedure on page 2-8)

	Command	Purpose
Step 3	**username** *name* [**password** [**0** | **5**] *password*] [expire date] [role *role-name*] **Example:** n1000v(config)# username NewUser password 4Ty18Rnt	Creates a user account. • **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	**show user-account** *username* **Example:** n1000v(config)# show user-account NewUser user:NewUser this user account has no expiry date roles:network-operator network-admin n1000v(config)#	Displays the new user account configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Creating a Role

Use this procedure to create a role defining a set of specific actions that are permitted or denied. This role will be assigned to users whose access requirements match the actions defined.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are 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 that 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.

SUMMARY STEPS

1. **config t**
2. **role name** *role-name*
3. (Optional) **description** *string*
4. **rule** *number* {**deny** | **permit**} **command** *command-string*

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

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

   rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name*
5. Repeat Step 4 to create all needed rules for this role.
6. **show role**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**role name** *role-name*
	
	Example:
	`n1000v(config)# role name UserA`
	`n1000v(config-role)#`
		Names a user role and places you in Role Configuration mode for that role.
		
		The name is a case-sensitive, alphanumeric string of up to 16 characters.

Step 3	**description** *description-string* **Example:** `n1000v(config-role)# description Prohibits use of clear commands`	(Optional) Configures the role description, which can include spaces.

	Command	Purpose
Step 4	**rule** *number* {**deny** | **permit**} **command** *command-string* **Example:** `n1000v(config-role)# rule 1 deny command clear users`	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/denies access to all Ethernet interfaces. This example rule denies access to the **clear users** command.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **Example:** `n1000v(config-role)# rule 2 deny read-write`	Creates a blanket rule to permit or deny all operations. This example rule permits read-only access for any operation.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name* **Example:** `n1000v(config-role)# rule 3 permit read feature eth-port-sec`	Creates a rule for feature access. Use the **show role feature** command to display a list of available features. This example rule permits users read-only access to the Ethernet port security feature.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name* **Example:** `n1000v(config-role)# rule 4 deny read-write feature-group eth-port-sec`	Creates a rule for feature group access. Use the **show role feature-group** command to display a list of feature groups. This example configures a rule denying access to a feature group.
Step 5	Repeat Step 4 to create all needed rules for the specified role.	
Step 6	**show role** **Example:** `n1000v(config)# show role`	(Optional) Displays the user role configuration.
Step 7	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Creating a Feature Group

Use this procedure to create and configure a feature group.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You can create up to 64 custom feature groups.

SUMMARY STEPS

1. **config t**
2. **role feature-group name** *group-name*
3. **show role feature**
4. **feature** *feature-name*
5. **Repeat 4 for all features to be added to the feature group.**
6. **show role feature-group**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**role feature-group name** *group-name*
	
	Example:
	`n1000v(config)# role feature-group name GroupA`
	`n1000v(config-role-featuregrp)#`
		Places you into the Role Feature Group Configuration mode for the named group.
		• **group-name**: A case-sensitive, alphanumeric string of up to 32 characters in length.

Step 3	**show role feature**
	
	Example:
	`n1000v(config-role-featuregrp)# show role feature`
	`feature: aaa`
	`feature: access-list`
	`feature: cdp`
	`feature: install`
	`. . .`
	`n1000v(config-role-featuregrp)#`
		Displays a list of available features for use in defining the feature group.

Step 4	**feature** *feature-name*
	
	Example:
	`n1000v(config-role-featuregrp)# feature syslog`
	`n1000v(config-role-featuregrp)#`
		Adds a feature to the feature group.

Step 5	Repeat Step 6 for all features to be added to the feature group.	

	Command	Purpose

Step 6	**show role feature-group**
	
	Example:
	`n1000v(config-role-featuregrp)# show role feature-group`
	`feature group: GroupA`
	`feature: syslog`
	`feature: snmp`
	`feature: ping`
	`n1000v(config-role-featuregrp)#`
		(Optional) Displays the feature group configuration.

Step 7	**copy running-config startup-config** **Example:** `n1000v(config-role-featuregrp)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring Interface Access

Use this procedure to configure interface access for a specific role.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already created one or more user roles using the "Creating a Role" procedure on page 2-8. In this procedure, you will be modifying a role you have already created.
- By default, a role allows access to all interfaces. In this procedure you will, first, deny access to all interfaces and then permit access to selected interfaces.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into the CLI Global Configuration mode.
Step 2	**role name** *role-name* **Example:** n1000v(config)# role name network-observer n1000v(config-role)#	Specifies a user role and enters Role Configuration mode for the named role.
Step 3	**interface policy deny** **Example:** n1000v(config-role)# interface policy deny n1000v(config-role-interface)#	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	**permit interface** *interface-list* **Example:** n1000v(config-role-interface)# permit interface ethernet 2/1-4	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	**show role** *role-name* **Example:** n1000v(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	(Optional) Displays the role configuration.
Step 6	**copy running-config startup-config** **Example:** n1000v(config-role-featuregrp)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring VLAN Access

Use this procedure to define the VLAN access for a role.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already created one or more user roles using the "Creating a Role" procedure on page 2-8. In this procedure, you will be modifying a role you have already created.

- By default, access is allowed to all VLANs. In this procedure you will, first, deny access to all VLANs and then permit access to selected VLANs.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**role name** *role-name*
	
	Example:
	`n1000v(config)# role name network-observer`
	`n1000v(config-role)#`
		Specifies a user role and enters role configuration mode.

Step 3	**vlan policy deny**
	
	Example:
	`n1000v(config-role)# vlan policy deny`
	`n1000v(config-role-vlan)#`
		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	**permit vlan** *vlan-list* **Example:** `n1000v(config-role-vlan)# permit vlan 1-4`	Specifies the VLAN(s) that users assigned to this role can access. Repeat this command to specify all VLANs that users assigned to this role are permitted to access.

	Command	Purpose

Step 5	**show role** *role-name*
	
	Example:
	`n1000v(config-role)# show role network-observer`
	
	`role: network-observer`
	`  description: temp`
	`  Vlan policy: deny`
	`  Permitted vlans: vlan 1-4`
	`  Interface policy: deny`
	`  Permitted interfaces: Ethernet2/1-4`
		(Optional) Displays the role configuration.

Step 6	**copy running-config startup-config** **Example:** `n1000v(config-role)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying the User Access Configuration

To display user account and RBAC configuration information, perform one of the following tasks:

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

Example Configuration

The following example shows how to configure a role:

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

The following example shows how to configure a feature group:

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

Additional References

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

- Related Documents, page 2-16
- Standards, page 2-16
- MIBs, page 2-16

Related Documents

Related Topic	Document Title
User access commands	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*
Managing users on the switch	*Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)*

Standards

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

MIBs

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

Feature History for User Accounts

This section provides the user accounts release history.

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

CHAPTER 3

Configuring VSD

This chapter describes how to configure VSD and includes the following topics:

Information About Virtual Service Domain

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

Service Virtual Machine

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

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

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

Port Profiles

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

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

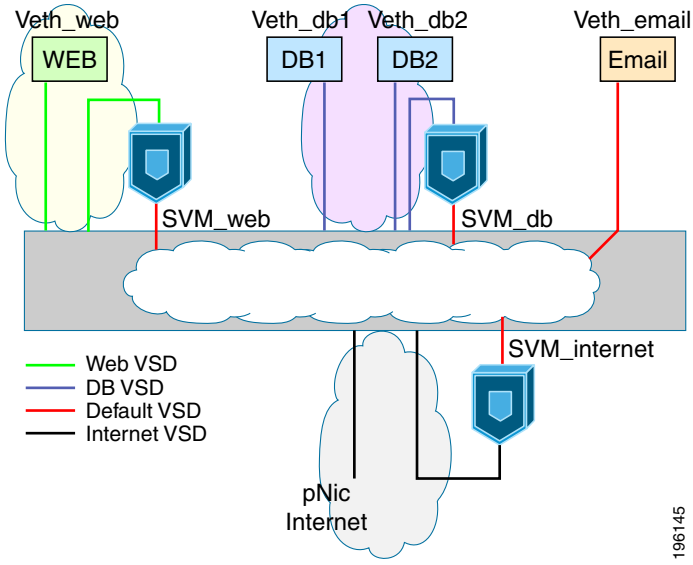
A VSD is formed by creating the following port profiles:

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

In Figure 3-1, a single VEM takes the place of vswitches; the SVMs define the following VSDs;

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

Figure 3-1 Virtual Service Domain (VSD) Example



Guidelines and Limitations

Virtual Service Domain has the following configuration guidelines and limitations:

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

Default Settings

The following table lists the Telnet defaults.

Parameters	Default
service-port default-action	Forward.
switchport trunk allowed vlan	All

Configuring VSD

This section includes the following procedures:

Configuring an Inside or Outside VSD Port Profile

Use this procedure to configure the port-profiles that define the connections going into and out of the SVM.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have taken the SVM out of service to prevent any configuration errors from flooding the network. Once the configuration is complete and verified, you can bring the SVM back into service.
- If you do not configure a service-port, the SVM will come up as a regular VM, flooding the network with packets.
- Selected VLAN filtering is not supported in this configuration. The default should be used instead, which allows all VLANs on the port.

SUMMARY STEPS

1. **config t**
2. **port-profile** *name*
3. **switchport mode trunk**
4. **switchport trunk allowed vlan** *vlanID*
5. **virtual-service-domain** *name*
6. **no shut**
7. **vmware port-group** *pg-name*
8. **service-port** {**inside** | **outside**} [**default-action** {**drop** | **forward**}]
9. **state enabled**
10. **show virtual-service-domain** *name*
11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**port-profile** *name*
	
	Example:
	`n1000v(config)# port-profile webserver-inside`
	`n1000v(config-port-profile)#`
		Creates a port profile and places you into Port Profile Configuration mode for the named port profile.
		
		The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.

Step 3	**switchport mode trunk**
	
	Example:
	`n1000v(config-port-profile)# switchport mode trunk`
	`n1000v(config-port-profile)#`
		Designates that the interfaces are switch trunk ports.

Step 4	**switchport trunk allowed vlan** *vlanID*
	
	Example:
	`n1000v(config-port-profile)# switchport trunk allowed vlan all`
	`n1000v(config-port-profile)#`
		Allows all VLANs on the port.

Step 5	**virtual-service-domain** *name*
	
	Example:
	`n1000v(config-port-profile)# virtual-service-domain vsd1-webserver`
	`n1000v(config-port-profile)#`
		Adds a VSD name to this port profile.

Step 6	**no shutdown**
	
	Example:
	`n1000v(config-port-prof)# no shutdown`
	`n1000v(config-port-prof)#`
		Administratively enables all ports in the profile.

Step 7	**vmware port-group** *pg-name*
	
	Example:
	`n1000v(config-port-prof)# vmware port-group webservers-inside-protected`
	`n1000v(config-port-prof)#`
		Designates the port-profile as a VMware port-group.
		
		The port profile is mapped to a VMware port group of the same name. When a vCenter Server connection is established, the port group created in Cisco Nexus 1000V is then distributed to the virtual switch on the vCenter Server.
		
		name: Port group name. If you do not specify a pg-name, then the port group name will be the same as the port profile name. If you want to map the port profile to a different port group name, use the pg-name option followed by the alternate name.

	Command	Purpose
Step 8	**service-port** {**inside** | **outside**} [**default-action** {**drop** | **forward**}]	Configures the interface as either inside or outside and designates (default-action) whether packets should be forwarded or dropped if the service port is down. If you do not specify a default-action, then the **forward** setting is used by default. **Caution** If you do not configure a service-port, the SVM will come up as a regular VM, flooding the network with packets.

	Example:
	`n1000v(config-port-prof)# service-port inside default-action forward`
	`n1000v(config-port-prof)#`
		This example configures an inside VSD that forwards packets if the service port is down.

	Example:
	`n1000v(config-port-prof)# service-port outside default-action forward`
	`n1000v(config-port-prof)#`
		This example configures an outside VSD that forwards packets if the service port is down.

Step 9	**state enabled**
	Example:
	`n1000v(config-port-prof)# state enabled`
	`n1000v(config-port-prof)#`
		Enables the VSD port profile.
		The configuration for this port profile is applied to the assigned ports, and the port group is created in the VMware vSwitch on the vCenter Server.

Step 10	**show virtual-service-domain** *name*
	Example:
	`n1000v(config-port-prof)# show virtual-service-domain vsd1-webserver`
	`Default Action: forward`
	`_____`
	`Interface Type`
	`_____`
	`Vethernet1 Member`
	`Vethernet2 Member`
	`Vethernet3 Member`
	`Vethernet7 Inside`
	`Vethernet8 Outside`
	`n1000v(config-port-prof)#`
		(Optional) Displays the configuration for this VSD port profile. Use this to verify that the port-profile was configured as expected.

Step 11	**copy running-config startup-config**
	Example:
	`n1000v(config-port-prof)# copy running-config startup-config`
	`[########################################] 100%`
	`n1000v(config-port-prof)#`
		(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring a Member VSD Port Profile

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Do not configure a member VSD port profile on an SVM.

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

SUMMARY STEPS

1. **config t**
2. **port-profile** *name*
3. **switchport access vlan** *vlanID*
4. **switchport trunk allowed vlan** *vlanID*
5. **virtual-service-domain** *name*
6. **no shut**
7. **state enabled**
8. **show virtual-service-domain** *name*
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 1	**port-profile** *name*
	
	Example:
	`n1000v(config)# port-profile vsd1-member`
	`n1000v(config-port-profile)#`
		Creates a port profile and places you into Port Profile Configuration mode for the named port profile.
		
		The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.

Step 2	**switchport access vlan** *vlanID*
	
	Example:
	`n1000v(config-port-profile)# switchport access vlan 315`
	`n1000v(config-port-profile)#`
		Assigns a VLAN ID to the access port for this port profile.

	Command	Purpose

Step 3	**virtual-service-domain** *name*
	Example:
	`n1000v(config-port-profile)# virtual-service-domain vsd1-webserver`
	`n1000v(config-port-profile)#`
		Assigns a VSD name to this port profile.

Step 4	**no shutdown**
	Example:
	`n1000v(config-port-prof)# no shutdown`
	`n1000v(config-port-prof)#`
		Administratively enables all ports in the profile.

Step 5	**state enabled**
	Example:
	`n1000v(config-port-prof)# state enabled`
	`n1000v(config-port-prof)#`
		Enables the VSD port profile.
		The configuration for this port profile is applied to the assigned ports, and the port group is created in the VMware vSwitch on the vCenter Server.

Step 6	**show virtual-service-domain** *name*
	Example:
	`n1000v(config-port-prof)# show virtual-service-domain vsd1-webserver`
	`Default Action: forward`
	`_____`
	`Interface Type`
	`_____`
	`Vethernet1 Member`
	`Vethernet2 Member`
	`Vethernet3 Member`
	`Vethernet6 Member`
	`Vethernet7 Inside`
	`Vethernet8 Outside`
	`n1000v(config-port-prof)#`
		(Optional) Displays the configuration for this VSD port profile. Use this to verify that the port-profile was configured as expected.

Step 7	**copy running-config startup-config**
	Example:
	`n1000v(config-port-prof)# copy running-config startup-config`
	`[########################################] 100%`
	`n1000v(config-port-prof)#`
		(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying the Configuration

To display the VSD configuration, use the following commands:

Command	Purpose
show virtual-service-domain name *vsd-name*	Displays a specific VSD configuration. See Example 3-1 on page 3-9.
show virtual-service-domain brief	Displays a summary of all VSD configurations. See Example 3-2 on page 3-9.

Command	Purpose
show virtual-service-domain interface	Displays the interface configuration for all VSDs. See Example 3-3 on page 3-9.
module vem *module_number* **execute vemcmd show vsd**	Displays the VEM VSD configuration by sending the command to the VEM from the remote Cisco Nexus 1000V. See Example 3-4 on page 3-10.
module vem *module_number* **execute vemcmd show vsd ports**	Displays the VEM VSD ports configuration by sending the command to the VEM from the remote Cisco Nexus 1000V. See Example 3-5 on page 3-10.

For detailed information about command output for these commands, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

***Example 3-1 show virtual-service-domain name** vsd_name*

```
n1000v## show virtual-service-domain name vsd1
Default Action: drop
_____________________________
Interface         Type
_____________________________
Vethernet1        Member
Vethernet2        Member
Vethernet3        Member
Vethernet6        Member
Vethernet7        Inside
Vethernet8        Outside

n1000v#
```

Example 3-2 show virtual-service-domain brief

```
n1000v# show virtual-service-domain brief
Name   vsd-id    default action    in-ports   out-ports   mem-ports    Modules with
                                                                      VSD Enabled
zone    1         forward           1          1           2            4
n1000v#
```

Example 3-3 show virtual-service-domain interface

```
n1000v# sho virtual-service-domain interface
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Name             Interface             Type      Status
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
vsd1             Vethernet1            Member    Active
vsd1             Vethernet2            Member    Active
vsd1             Vethernet3            Member    Active
vsd1             Vethernet6            Member    Active
vsd1             Vethernet7            Inside    Active
vsd1             Vethernet8            Outside   Active
vsd2             Vethernet9            Inside    Active
vsd2             Vethernet10           Outside   Active
```

Example 3-4 module** module_number **execute vemcmd show vsd

```
n1000v# module vem 4 execute vemcmd show vsd
ID Def_Act ILTL OLTL NMLTL State Member LTLs
1 FRWD 51 50 1 ENA 49
n1000v#
```

Example 3-5 module** module_number **execute vemcmd show vsd ports

```
n1000v# module vem 4 execute vemcmd show vsd ports
LTL IfIndex VSD_ID VSD_PORT_TYPE
49 1c000010 1 REGULAR
50 1c000040 1 OUTSIDE
51 1c000030 1 INSIDE
n1000v#
```

Configuration Example

The following example shows how to configure VSD.

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

Additional References

For additional information related to VSD configuration, see the following:

- Related Documents, page 3-11
- Standards, page 3-11

Related Documents

Related Topic	Document Title
Port Profiles	*Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*
CLI	*Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)* *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History

This section provides the VSD release history.

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

CHAPTER 4

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) and includes the following sections:

Information About AAA

This section includes the following topics:

AAA Security Services

Based on a user ID and password combination, AAA is used to authenticate and authorize users. A key secures communication with AAA servers.

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

Table 4-1 shows the related CLI command for configuring an AAA service.

Table 4-1 AAA Service Configuration Commands

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

AAA secures the following:

- Authentication, page 4-2
- Authorization, page 4-3
- Accounting, page 4-3

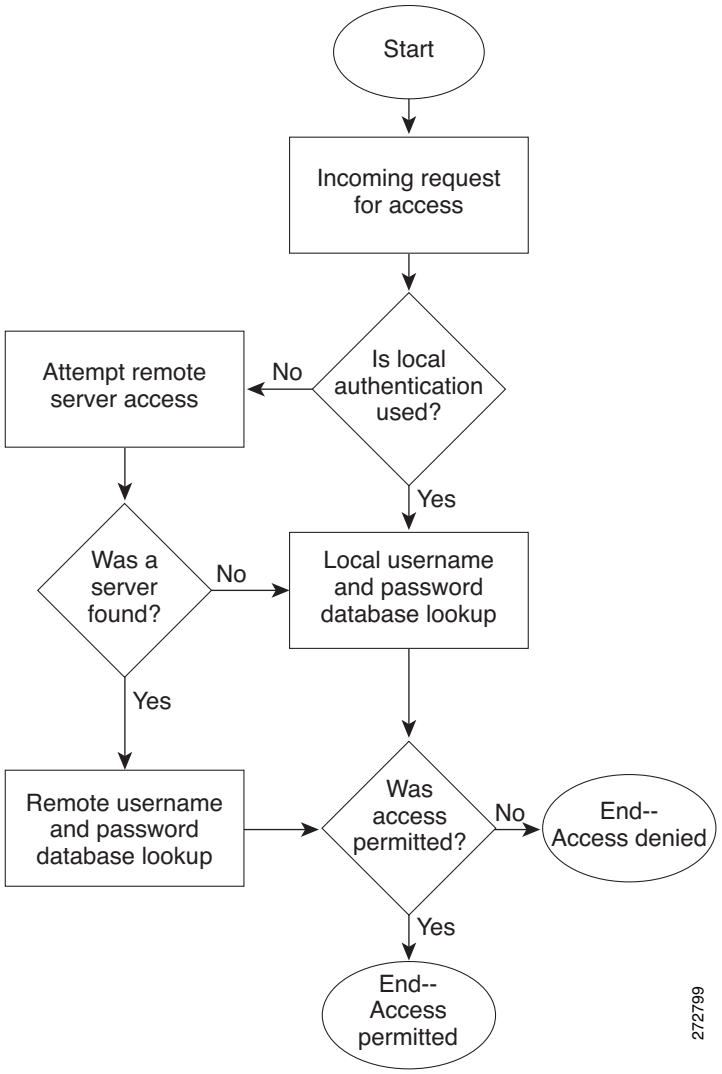
Authentication

Authentication identifies users with a login and password, messaging, and encryption.

Authentication is accomplished as follows:

Authentication Method	Description
Local database	Authenticates the following with a local lookup database of user names or passwords. • Console login authentication • User login authentication • User management session accounting
Remote RADIUS or TACACS+ server	Authenticates the following using a remote server lookup database of user names and 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

Figure 4-1 Authenticating User Log In



Authorization

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

Accounting

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 fail-over in case one remote AAA 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 fail-over for each other in this same way.

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

Prerequisites for AAA

Authentication using remote AAA servers requires that the following be in place:

- 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.

  See the "Configuring Shared Keys" procedure on page 6-9.

AAA Guidelines and Limitations

The Cisco Nexus 1000Vdoes not support user names made up of all numeric characters and does not create local user names made up of all numeric characters. If a username made up of all numeric characters exists on an AAA server and is entered during login, the Cisco Nexus 1000V does authenticate the user.

Default Settings

The following table lists the AAA defaults.

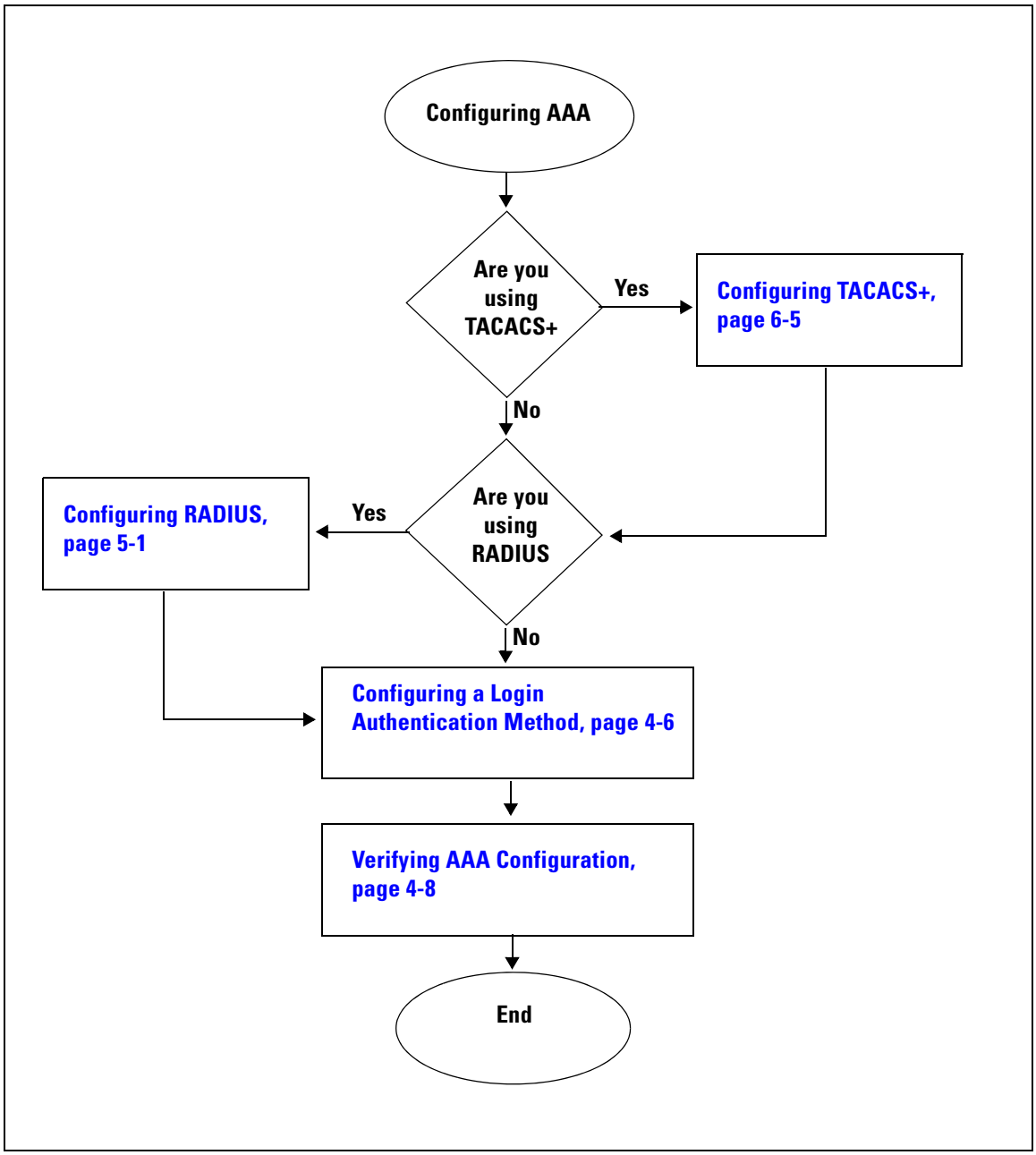
Parameters	Default
Console authentication method	**local**
Default authentication method	**local**
Login authentication failure messages	Disabled

Configuring AAA

This section includes the following topics:

Use the following flow chart to configure AAA.

Flow Chart: Configuring AAA



Configuring a Login Authentication Method

Use this procedure to configure the login authentication method.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- If authentication is to be done with TACACS+ server group(s), you have already added the group(s). For more information, see Configuring a TACACS+ Server Group, page 6-12.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**aaa authentication login {console | default}** {**group** *group-list* [**none**]| **local** | **none**} **Example:** `n1000v(config)# aaa authentication login console group tacgroup`	Configures the console or default login authentication method. • **group**: Authentication is done by server group(s). – **group-list**: List server group names separated by spaces; or none for 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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

	Command	Purpose

Step 4	**show aaa authentication**
	
	Example:
	`n1000v# show aaa authentication`
	`default: group tacgroup`
	`console: group tacgroup`
	`n1000v#`
		(Optional) Displays the configured login authentication method.

Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

Use this procedure to enable the login authentication failure message to displays if the remote AAA servers do not respond.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The following is the Login Authentication Failure message:

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

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in CLI Global Configuration mode.

Step 2	**aaa authentication login error-enable**
	
	Example:
	`n1000v(config)# aaa authentication login error-enable`
	`n1000v(config)#`
		Enables login authentication failure messages. The default is disabled.

	Command	Purpose

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits CLI Global Configuration mode and returns you to EXEC mode.

Step 4	**show aaa authentication login error-enable**
	
	Example:
	
	`n1000v# show aaa authentication login error-enable`
	`enabled`
	`n1000v#`
		(Optional) Displays the login failure message configuration.

Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying AAA Configuration

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

Command	Purpose
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information. See Example 4-1 on page 4-8
show aaa groups	Displays the AAA server group configuration.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration. See Example 4-2 on page 4-8
show startup-config aaa	Displays the AAA configuration in the startup configuration. See Example 4-3 on page 4-9

Example 4-1 show aaa authentication

```
n1000v# show aaa authentication login error-enable
disabled
```

Example 4-2 show running config aaa

```
n1000v# 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
n1000v#
```

Example 4-3 show startup-config aaa

```
n1000v# show startup-config aaa
version 4.0(1)svs#
```

Example AAA Configuration

The following is an AAA configuration example:

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

Additional References

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

- Related Documents, page 4-9
- Standards, page 4-9

Related Documents

Related Topic	Document Title
System Management	*Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(4)*
CLI	*Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)*
TACACS+ Security protocol	Chapter 6, “Configuring TACACS+”

Standards

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

Feature History for AAA

This section provides the AAA release history.

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

CHAPTER 5

Configuring RADIUS

This chapter describes how to configure RADIUS protocol on Cisco NX-OS devices.

This chapter includes the following sections:

Information About RADIUS

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

This section includes the following topics:

RADIUS Network Environments

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

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

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

RADIUS Operation

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

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:

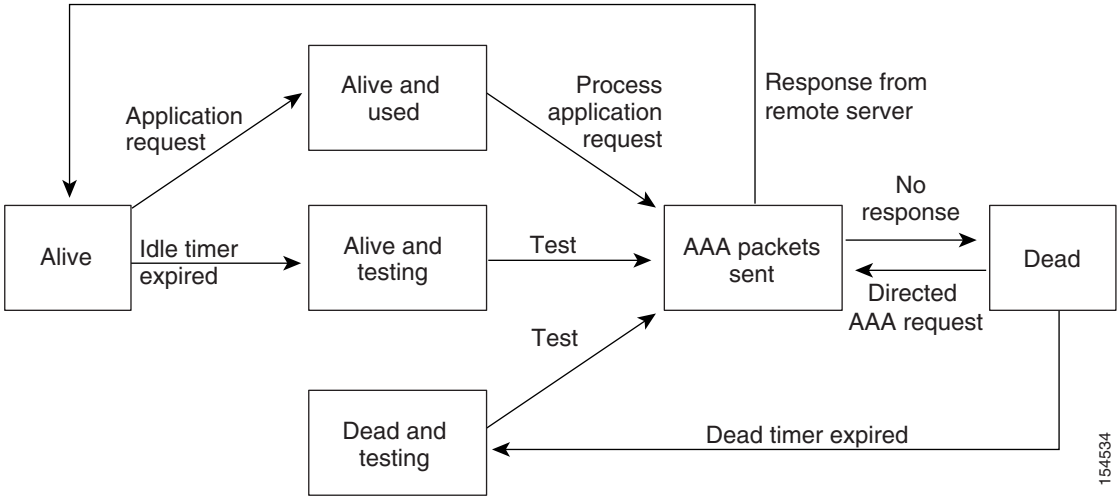
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

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. See Figure 5-1.

Figure 5-1 RADIUS Server States



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

Vendor-Specific Attributes

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

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

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

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

The following are supported VSA protocol options:

- 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 are supported attributes:

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

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

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

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

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

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

  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

RADIUS has the following prerequisites:

- 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

RADIUS has the following guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers.

Default Settings

Table 5-1 lists the RADIUS default settings.

Table 5-1 Default RADIUS Parameters

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

Configuring RADIUS Servers

This section includes the following topics:

Note Be aware that the Cisco NX-OS commands for this feature may differ from those used in Cisco IOS.

Configuring RADIUS Server Hosts

Use this procedure to configure the IP address or the hostname for each RADIUS server to be used for authentication.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You can configure up to 64 RADIUS servers.
- All RADIUS server hosts are automatically added to the default RADIUS server group.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **Example:** `n1000v(config)# radius-server host 10.10.1.1`	Defines the IP address or hostname for the RADIUS server.

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

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

SUMMARY STEPS

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

DETAILED STEPS

To configure a global preshared key, follow these steps:

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server key** [**0** | **7**] *key-value* **Example:** `n1000v(config)# radius-server key 0 QsEfThUkO`	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

Configuring a RADIUS Server Key

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have the key to be used for the remote RADIUS host.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value* **Example:** `n1000v(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

Configuring RADIUS Server Groups

Use this procedure to configure a RADIUS server group whose member servers share authentication functions.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- All servers in a RADIUS server group must belong to the RADIUS protocol.
- The servers in the group are tried in the same order in which you configure them.

SUMMARY STEPS

1. **config t**
2. **aaa group server radius** *group-name*
3. **server** {*ipv4-address* | *server-name*}
4. **deadtime** *minutes*
5. **use-vrf** *vrf-name*
6. (Optional) **source-interface** {*interface-type*} {*interface-number*}
7. (Optional) **show radius-server groups** [*group-name*]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**aaa group server radius** *group-name*
	
	Example:
	`n1000v(config)# aaa group server radius RadServer`
	`n1000v(config-radius)#`
		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	**server** {*ipv4-address* | *server-name*} **Example:** `n1000v(config-radius)# server 10.10.1.1`	Configures the RADIUS server as a member of the RADIUS server group. **Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.

	Command	Purpose
Step 4	**deadtime** *minutes* **Example:** `n1000v(config-radius)# deadtime 30`	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value (see the "Configuring the Global Dead-Time Interval" section on page 5-20).
Step 5	**use-vrf** *vrf-name* **Example:** `n1000v(config-radius)# use-vrf vrf1`	(Optional) Specifies the VRF to use to contact the servers in the server group.

Step 6	**source-interface** {*interface-type*} {*interface-number*}
	
	Example:
	`n1000v(config-radius)# source-interface mgmt0`
	`n1000v(config-radius)#`
		(Optional) Specifies a source interface to be used to reach the RADIUS 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	**show radius-server groups** [*group-name*]	(Optional) Displays the RADIUS server group configuration.

Example:

```
n1000v(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
```

Step 8	**copy running-config startup-config** **Example:** `n1000v(config-radius)# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

Enabling RADIUS Server Directed Requests

Use this procedure to let users 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 VRF to use and *hostname* is the name of a configured RADIUS server.

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Directed requests are disabled by default.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into the CLI Global Configuration mode.
Step 2	n1000v(config)# radius-server directed-request **Example:** n1000v(config)# radius-server directed-request	Enables directed requests. The default is disabled.
Step 3	**exit** **Example:** n1000v(config)# exit n1000v#	Returns you to the CLI EXEC mode.
Step 4	**show radius-server directed-request** **Example:** n1000v# show radius-server directed-request	(Optional) Displays the directed request configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Setting the Global Timeout for All RADIUS Servers

Use this procedure to configure the global timeout interval specifying how long to wait for a response from a RADIUS server before declaring a timeout failure.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The timeout specified in the "Setting the Timeout Interval for a Single RADIUS Server" procedure on page 5-14 overrides the global RADIUS timeout.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server timeout** *seconds* **Example:** `n1000v(config)# radius-server timeout 10`	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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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" procedure on page 5-15, overrides this global setting.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server retransmit** *count* **Example:** `n1000v(config)# radius-server retransmit 3`	Defines the number of retransmits allowed before reverting to local authentication. This is a global setting that applies to all RADIUS servers. The default number of retransmits is 1 and the range is from 0 to 5.

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

	Command	Purpose
Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The timeout specified for a single RADIUS server overrides the timeout defined in the "Setting the Global Timeout for All RADIUS Servers" procedure on page 5-12.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **timeout** *seconds* **Example:** `n1000v(config)# radius-server host server1 timeout 10`	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.

	Command	Purpose

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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 is applied to a single RADIUS server and takes precedence over the global retry count.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count* **Example:** `n1000v(config)# radius-server host server1 retransmit 3`	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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

Configuring a RADIUS Accounting Server

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, RADIUS servers are used for both accounting and authentication.
- You know the destination UDP port number for RADIUS accounting messages.

SUMMARY STEPS

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

DETAILED STEPS

To configure the authentication and accounting attributes for RADIUS servers, follow these steps:

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into the CLI Global Configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *host-name*} **acct-port** *udp-port* **Example:** n1000v(config)# radius-server host 10.10.1.1 acct-port 2004	(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	**radius-server host** {*ipv4-address* | *host-name*} **accounting** **Example:** n1000v(config)# radius-server host 10.10.1.1 accounting	(Optional) Designates the specific RADIUS host as an accounting server. The default is both accounting and authentication.
Step 4	**exit** **Example:** n1000v(config)# exit n1000v#	Returns you to the CLI EXEC mode.
Step 5	**show radius-server** **Example:** n1000v(config)# show radius-server	(Optional) Displays the RADIUS server configuration.
Step 6	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Saves this change in the running configuration to the startup configuration.

Configuring a RADIUS Authentication Server

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, RADIUS servers are used for both accounting and authentication.
- You know the destination UDP port number for RADIUS authentication messages.

SUMMARY STEPS

1. **config t**
2. **radius-server host** {*ipv4-address* | *host-name*} **auth-port** *udp-port*
3. **radius-server host** {*ipv4-address* | *host-name*} **authentication**

4. **exit**
5. **show radius-server**
6. **copy running-config startup-config**

DETAILED STEPS

To configure the authentication and accounting attributes for RADIUS servers, follow these steps:

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **auth-port** *udp-port* **Example:** `n1000v(config)# radius-server host 10.10.2.2 auth-port 2005`	(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	**radius-server host** {*ipv4-address* | *host-name*} **authentication** **Example:** `n1000v(config)# radius-server host 10.10.2.2 authentication`	(Optional) Designates the specific RADIUS host as an authentication server. The default is both accounting and authentication.

Step 4	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 5	**show radius-server** **Example:** `n1000v(config)# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 6	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

Configuring Periodic RADIUS Server Monitoring

Use this procedure to configure the monitoring of RADIUS servers.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The test idle timer specifies the interval of time that elapses before a test packet is sent to a nonresponsive RADIUS server.

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `n1000v(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	**radius-server dead-time** *minutes* **Example:** `n1000v(config)# radius-server dead-time 5`	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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

	Command	Purpose
Step 5	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 6	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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 (see the "Configuring RADIUS Server Groups" section on page 5-9).

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

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

DETAILED STEPS

To configure the RADIUS dead-time interval, follow these steps:

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	`n1000v(config)#` **radius-server deadtime** *minutes* **Example:** `n1000v(config)# radius-server deadtime 5`	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.

	Command	Purpose

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Returns you to the CLI EXEC mode.

Step 4	**show radius-server** **Example:** `n1000v# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Saves this change in the running configuration to the startup configuration.

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 know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password* **Example:** `n1000v# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH`	Sends a test message to a RADIUS server to confirm availability.
Step 1	**test aaa group** *group-name username password* **Example:** `n1000v# test aaa group RadGroup user2 As3He3CI`	Sends a test message to a RADIUS server group to confirm availability.

Verifying RADIUS Configuration

Use the commands in this section to verify the RADIUS configuration. For detailed information about show command output, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

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 server activity.

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

Example RADIUS Configuration

The following example shows how to configure RADIUS:

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

Additional References

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

- Related Documents, page 5-22
- Standards, page 5-23

Related Documents

Related Topic	Document Title
Command reference	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for RADIUS

This section provides the RADIUS release history.

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

CHAPTER 6

Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol.

This chapter includes the following sections:

Information About TACACS+

You can use TACACS+ to provide centralized validation of users attempting to gain access to a device. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your device are available.

The TACACS+ security protocol provides centralized validation of users attempting to gain access to a device. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your 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.

This section includes the following topics:

TACACS+ Operation for User Login

THe following sequence of events take place when you attempt to login to a TACACS+ server using 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 may include prompts for additional information, such as 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

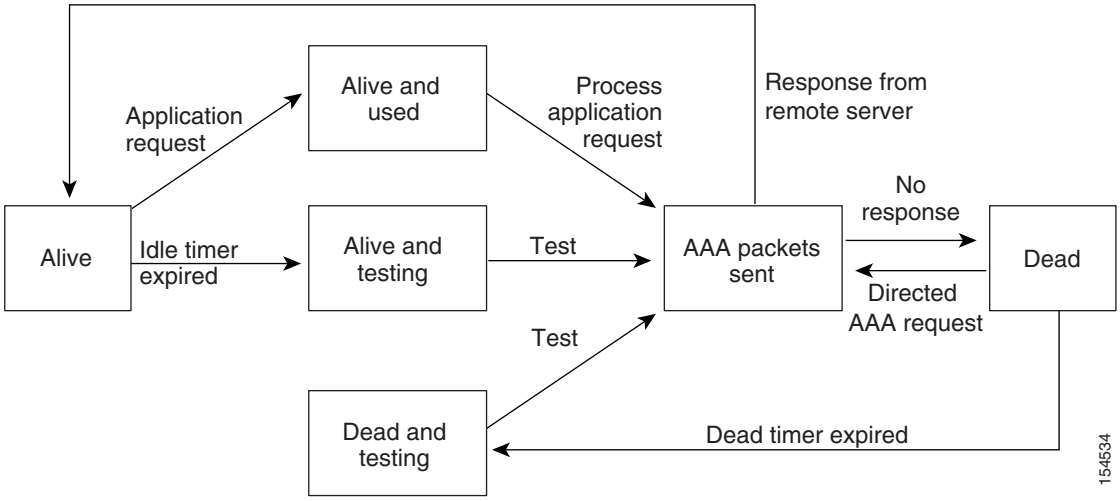
You must configure the TACACS+ preshared key to authenticate to the TACACS+ server. A preshared key is a secret text string shared between the device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not 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 and 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 6-1 TACACS+ Server States



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

Vendor-Specific Attributes

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

Cisco VSA Format

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

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

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

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

The following are supported VSA protocol options:

- 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 are other supported attributes:

- roles—Lists all the roles to which the user belongs. The value consists of a string listing 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+

TACACS+ has the following prerequisites:

- 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 using the following procedures:
  - Configuring a Login Authentication Method, page 4-6
  - Configuring AAA, page 4-4

Guidelines and Limitations

TACACS+ has the following guidelines and limitations:

- You can configure a maximum of 64 TACACS+ servers.
- The logging level for TACACS + must be set to 5.

Default Settings

The following table lists the TACACS+ defaults.

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes

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

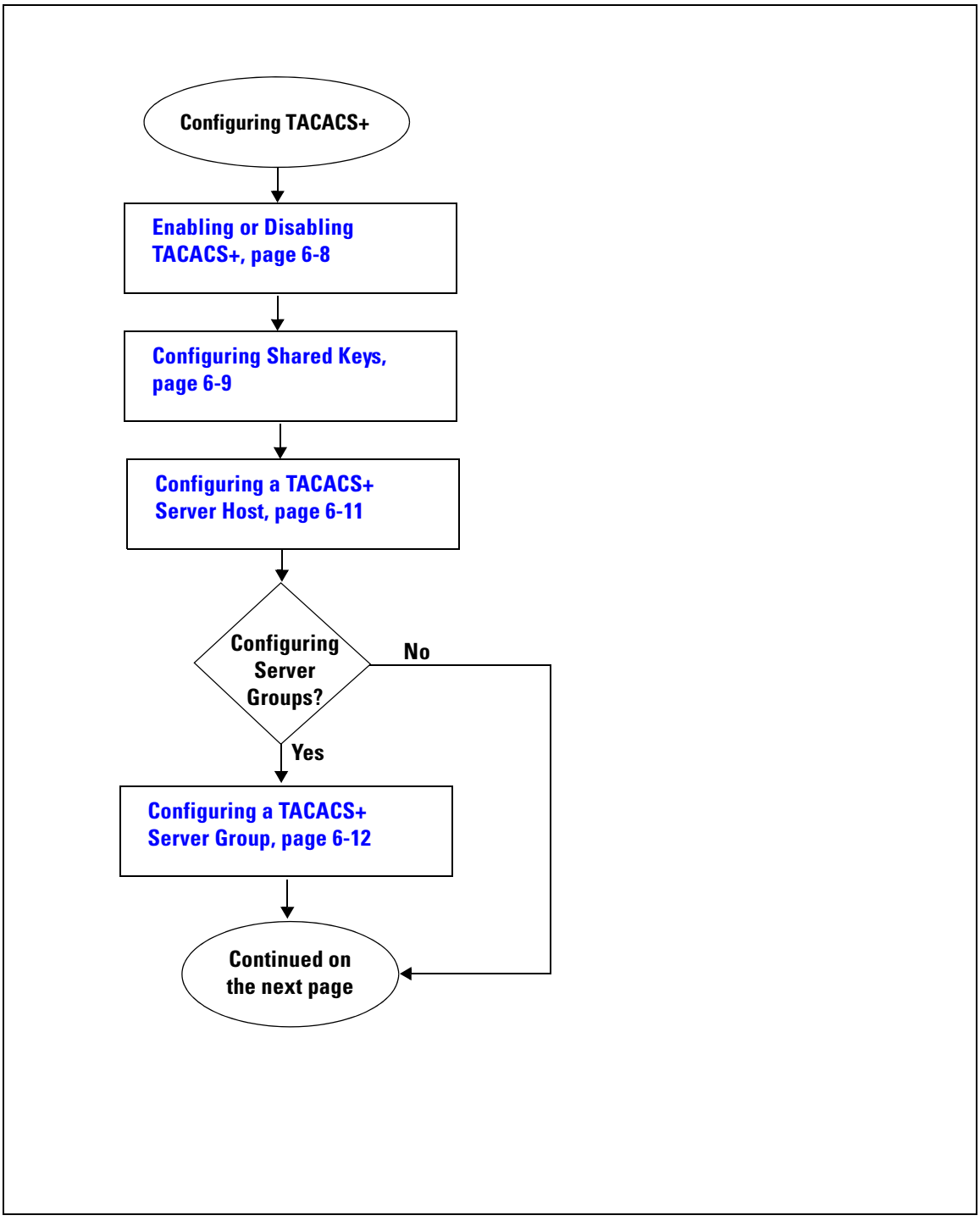
Configuring TACACS+

This section includes the following topics:

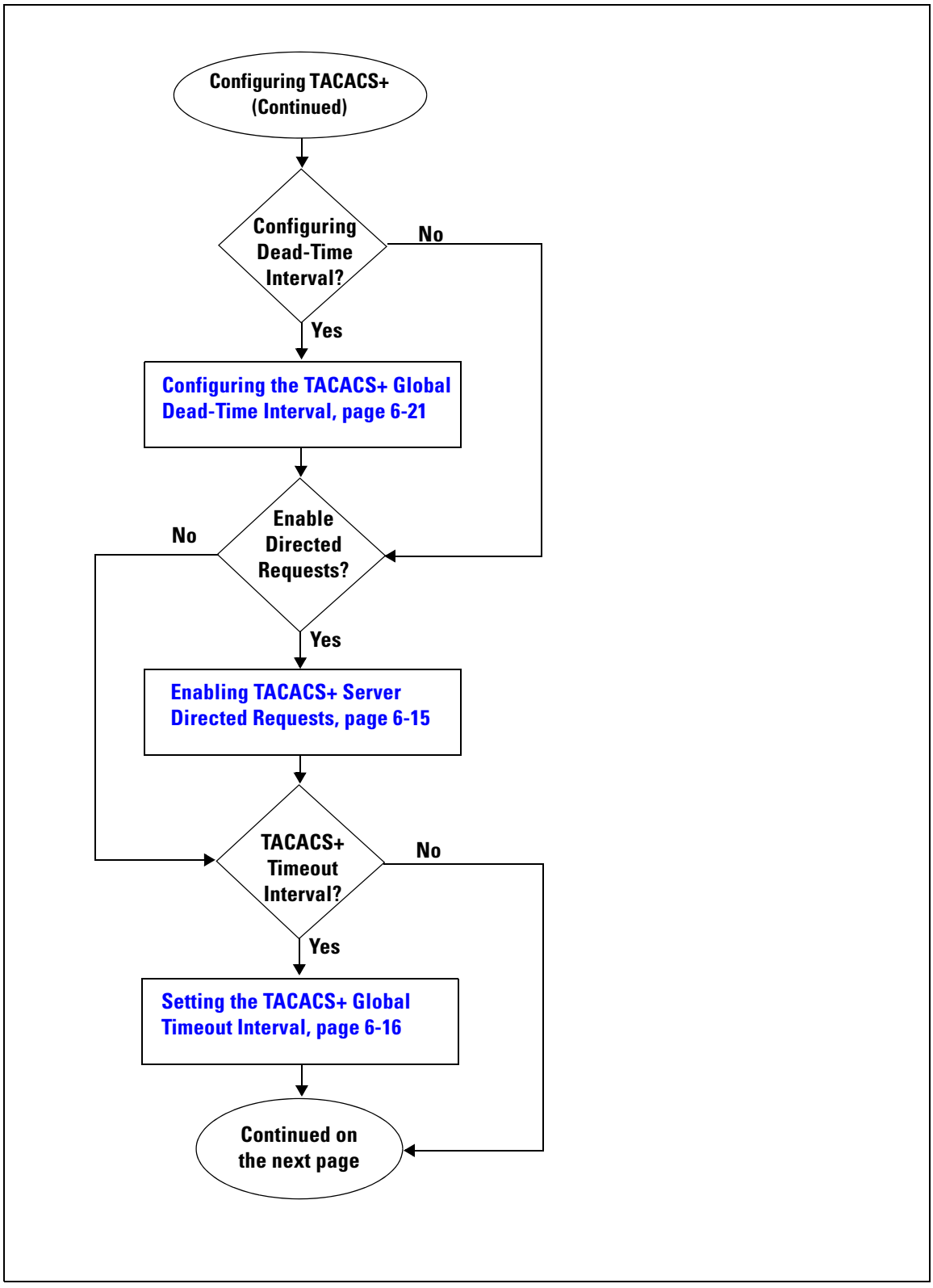
Note Be aware that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Use the following flow chart to configure TACACS+.

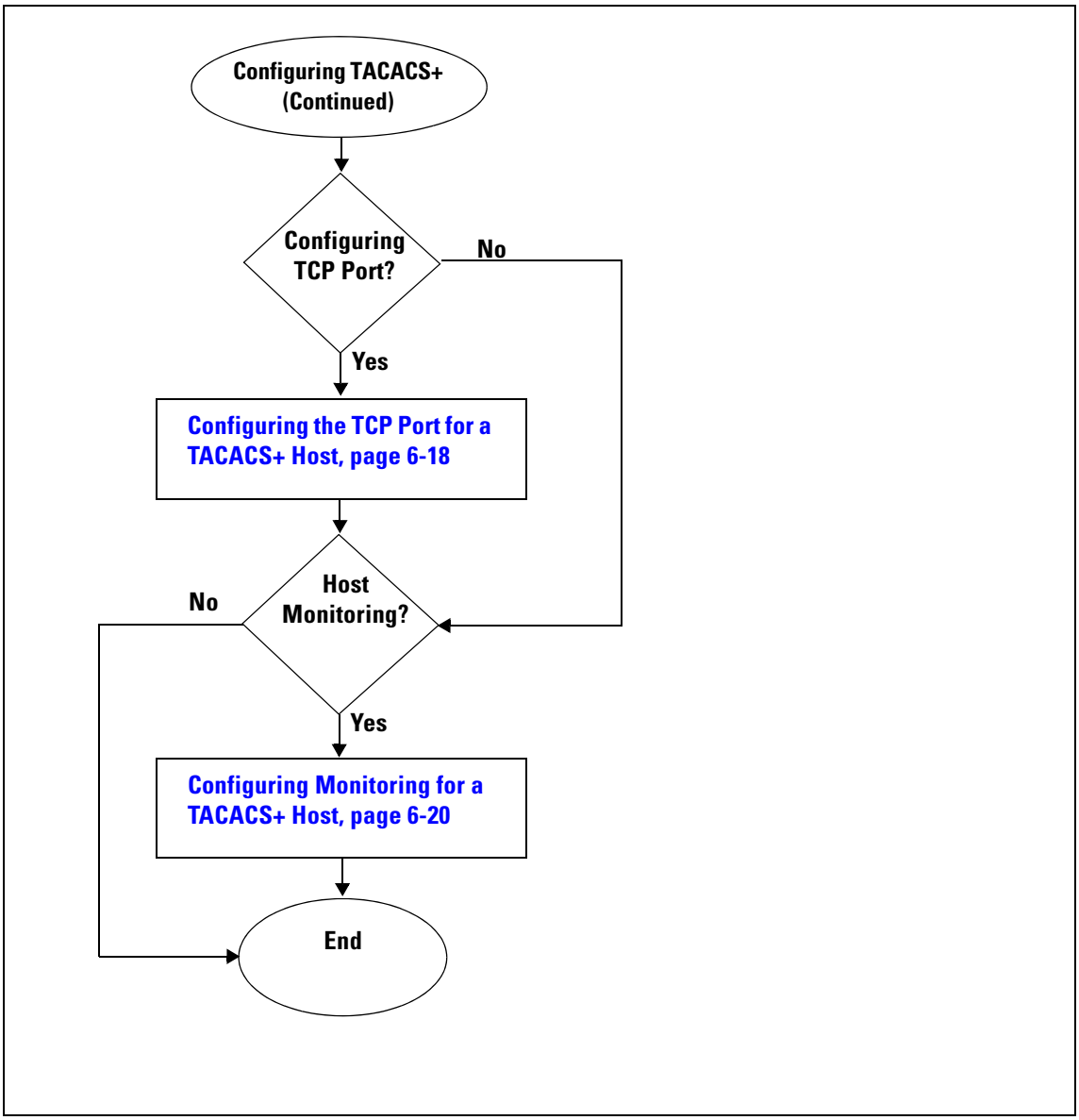
Flow Chart: Configuring TACACS+



Flow Chart: Configuring TACACS+ (Continued)



Flow Chart: Configuring TACACS+ (Continued)



Enabling or Disabling TACACS+

Use this procedure to either enable or disable TACACS+.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.
- 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**[no] tacacs+ enable** **Example:** n1000v(config)# tacacs+ enable n1000v(config)# **Example:** n1000v(config)# no tacacs+ enable n1000v(config)#	Enables or disables TACACS+.
Step 3	**exit** **Example:** n1000v(config)# exit n1000v#	Exits the CLI Global Configuration mode and returns you to EXEC mode.
Step 4	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies the changes you made to the startup configuration.

Configuring Shared Keys

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

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- You know the key for the TACACS+ server host(s).
- By default, no global key is configured.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	Do one of the following: • To configure a global key for all TACACS+ server hosts, continue with the next step. • To configure a key for a single TACACS+ server host, go to Step 5.	

Step 3	**tacacs-server key** [**0** | **7**] *global_key*
	
	Example:
	`n1000v(config)# tacacs-server key 0 QsEFtkI#`
	`n1000v(config)#`
		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]
		7: Specifies an encrypted string (key) to follow.
		global_key: A string of up to 63 characters.
		By default, no global key is configured.

Step 4	Go to Step 6.	

Step 5	**tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *shared_key*
	
	Example:
	`n1000v(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg`
	`n1000v(config)#`
		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.
		This shared key is used instead of the global shared key.

	Command	Purpose

Step 6	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

Step 7	**show tacacs-server**
	
	Example:
	`n1000v# 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`
		(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 8	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies these changes in the running configuration to the startup configuration.

Configuring a TACACS+ Server Host

Use this procedure to configure a TACACS+ server as a TACACS+ host.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- You have already configured the shared key, using the following:

  "Configuring Shared Keys" procedure on page 6-9
- You know the IP addresses or the hostnames for the remote TACACS+ server hosts.
- All TACACS+ server hosts are added to the default TACACS+ server group.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *host-name*} **Example:** n1000v(config)# tacacs-server host 10.10.2.2	Configures the server IP address or hostname as a TACACS+ server host.
Step 3	**exit** **Example:** n1000v(config)# exit n1000v#	Exits the CLI Global Configuration mode and returns you to EXEC mode.
Step 4	**show tacacs-server** **Example:** n1000v# **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 n1000v#	(Optional) Displays the TACACS+ server configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies these changes in the running configuration to the startup configuration.

Configuring a TACACS+ Server Group

Use this procedure to configure a TACACS+ server group whose member servers share authentication functions.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- All servers added to a TACACS+ server group must use the TACACS+ protocol.
- Once the TACACS+ server group is configured, the server members are tried in the same order in which you configured them.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.

- You have already configured the preshared keys, using the following:

  "Configuring Shared Keys" procedure on page 6-9

- A TACACS+ server group can provide fail-over in case 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 fail-over for each other in this same way.

SUMMARY STEPS

1. **config t**
2. **aaa group server tacacs+** *group-name*
3. **server** {*ipv4-address* | *host-name*}
4. **deadtime** *minutes*
5. **use-vrf** *vrf-name*
6. (Optional) **source-interface** {*interface-type*} {*interface-number*}
7. (Optional) **show tacacs-server groups**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**aaa group server tacacs+** *group-name*
	
	Example:
	`n1000v(config)# aaa group server tacacs+ TacServer`
	`n1000v(config-tacacs+)#`
		Creates a TACACS+ server group with the specified name and places you into the TACACS+ configuration mode for that group.

Step 3	**server** {*ipv4-address* | *host-name*}
	
	Example:
	`n1000v(config-tacacs+)# server 10.10.2.2`
	`n1000v(config-tacacs+)#`
		Configures the TACACS+ server host-name or IP address as a member of the TACACS+ server group.
		
		Tip If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.

Step 4	**deadtime** *minutes*
	
	Example:
	`n1000v(config-tacacs+)# deadtime 30`
	`n1000v(config-tacacs+)#`
		(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 (see the "Configuring the TACACS+ Global Dead-Time Interval" procedure on page 6-21).

	Command	Purpose

Step 5	**use-vrf** *vrf-name*
	
	Example:
	`n1000v(config-tacacs+)# use-vrf management`
	`n1000v(config-tacacs+)#`
		(Optional) Specifies the virtual routing and forwarding instance (VRF) to use to contact this server group.

Step 6	**source-interface** {*interface-type*} {*interface-number*}
	
	Example:
	`n1000v(config-tacacs+)# source-interface mgmt0`
	`n1000v(config-tacacs+)#`
		(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	**show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration.

Example:

```
n1000v(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
n1000v(config-tacacs+)#
```

Step 8	**copy running-config startup-config** **Example:** `n1000v(config-tacacs+)# copy running-config startup-config`	(Optional) Copies these changes made in the running configuration to the startup configuration.

Example:

```
n1000v(config)# aaa group server tacacs+ TacServer
n1000v(config-tacacs+)# server 10.10.2.2
n1000v(config-tacacs+)# deadtime 30
n1000v(config-tacacs+)# use-vrf management
n1000v(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
n1000v(config-tacacs+)#
```

Enabling TACACS+ Server Directed Requests

Use this procedure to let users designate the TACACS+ server to send their authentication request to. This is called a directed-request.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.

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

- 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**tacacs-server directed-request**
	
	Example:
	`n1000v(config)# tacacs-server directed-request`
	`n1000v(config)#`
		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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

	Command	Purpose

Step 4	**show tacacs-server directed-request**
	
	Example:
	`n1000v# show tacacs-server directed-request`
	`enabled`
	`n1000v#`
		(Optional) Displays the TACACS+ directed request configuration.

Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- The timeout specified for an individual TACACS+ server overrides the global timeout interval. To set the timeout for an individual server, see the "Setting a Timeout Interval for an Individual TACACS+ Host" procedure on page 6-17.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**tacacs-server timeout** *seconds* **Example:** n1000v(config)# tacacs-server timeout 10	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	**exit** **Example:** n1000v(config)# exit n1000v#	Exits the CLI Global Configuration mode and returns you to EXEC mode.
Step 4	**show tacacs-server** **Example:** 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 n1000v#	(Optional) Displays the TACACS+ server configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies these changes made in the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- The timeout setting for an individual TACACS+ server overrides the global timeout interval.

SUMMARY STEPS

1. **config t**
2. **tacacs-server host** {*ipv4-address* | *host-name*} **timeout** *seconds*

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *host-name*} **timeout** *seconds* **Example:** n1000v(config)# tacacs-server host 10.10.2.2 timeout 10 n1000v(config)#	Specifies the timeout interval for a specific server. The default is the global timeout interval. For more information, see the "Setting the TACACS+ Global Timeout Interval" procedure on page 6-16.
Step 3	**exit** **Example:** n1000v(config)# exit n1000v#	Exits the CLI Global Configuration mode and returns you to EXEC mode.
Step 4	**show tacacs-server** **Example:** 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 n1000v#	(Optional) Displays the TACACS+ server configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies these changes made in the running configuration to the startup configuration.

Configuring the TCP Port for a TACACS+ Host

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.

- You have configured the TACACS+ server using the "Configuring a TACACS+ Server Host" procedure on page 6-11.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *host-name*} **port** *tcp-port*
	
	Example:
	`n1000v(config)# tacacs-server host 10.10.2.2 port 2`
	`n1000v(config)#`
		Specifies the TCP port to use.
		allowable range: 1 to 65535
		default: 49

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

Step 4	**show tacacs-server**
	
	Example:
	`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:2`
	`timeout:10`
	`n1000v#`
		(Optional) Displays the TACACS+ server configuration.

Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Monitoring for a TACACS+ Host

Use this procedure to configure periodic monitoring of a TACACS+ host.

BEFORE YOU BEGIN

Before starting this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- You have configured the TACACS+ server.
  See the "Configuring a TACACS+ Server Host" procedure on page 6-11.
- 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `n1000v(config)# tacacs-server host 10.10.2.2 test username pvk2 password a3z9yjqz7 idle-time 3`	Configures server monitoring. **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.

	Command	Purpose
Step 3	**tacacs-server dead-time** *minutes* **Example:** `n1000v(config)# tacacs-server dead-time 5`	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	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

Step 5	**show tacacs-server**
	
	Example:
	`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:2`
	`timeout:10`
	`n1000v#`
		(Optional) Displays the TACACS+ server configuration.

Step 6	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies these changes made to the running configuration to the startup configuration.

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.

BEFORE YOU BEGIN

Before starting this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- You have configured the TACACS+ server.
  See the "Configuring a TACACS+ Server Host" procedure on page 6-11.
- When the dead-timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per group (see the "Configuring a TACACS+ Server Group" procedure on page 6-12).

SUMMARY STEPS

1. **config t**
2. **tacacs-server deadtime** *minutes*
3. **exit**

4. **show tacacs-server**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 2	**tacacs-server deadtime** *minutes* **Example:** `n1000v(config)# tacacs-server deadtime 5`	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes

Step 3	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits the CLI Global Configuration mode and returns you to EXEC mode.

Step 4	**show tacacs-server** **Example:** `n1000v# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Displaying Statistics for a TACACS+ Host

Use this procedure to display the statistics for TACACS+ host.

BEFORE YOU BEGIN

Before starting this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already enabled TACACS+ for authentication.
  See the "Enabling or Disabling TACACS+" procedure on page 6-8.
- You have configured the TACACS+ server.
  See the "Configuring a TACACS+ Server Host" procedure on page 6-11.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**show tacacs-server statistics** {*hostname* | *ipv4-address*}	Displays statistics for a TACACS+ host.

Example:

```
n1000v# show tacacs-server statistics 10.10.1.1
Server is not monitored

Authentication Statistics
        failed transactions: 9
        sucessfull transactions: 2
        requests sent: 2
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0

Authorization Statistics
        failed transactions: 1
        sucessfull transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0

Accounting Statistics
        failed transactions: 0
        sucessfull transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0
```

Example TACACS+ Configuration

The following example shows a TACACS+ configuration:

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

Feature History for TACACS+

This section provides the TACACS+ release history.

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

Additional References

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

- Related Documents, page 6-24
- Standards, page 6-24

Related Documents

Related Topic	Document Title
CLI	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*
System Management	*Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(4)*

Standards

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

CHAPTER 7

Configuring SSH

This chapter describes how to configure Secure Shell Protocol (SSH).

This chapter includes the following sections:

Information About SSH

This section includes the following topics:

SSH Server

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

TACACS+ user authentication and locally stored user names and passwords is 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 using 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

SSH has the following prerequisite:

- You have configured 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

- 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

The following table lists the default settings for SSH.

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

Configuring SSH

This section includes the following topics:

Generating SSH Server Keys

Use this procedure to generate an SSH server key based on your security requirements.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The default SSH server key is an RSA key that is generated using 1024 bits.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**no feature ssh** **Example:** n1000v(config)# no feature ssh	Disables SSH.
Step 3	**ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]} **Example:** n1000v(config)# ssh key dsa 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	**feature ssh** **Example:** n1000v(config)# feature ssh	Enables SSH.
Step 5	**show ssh key** **Example:** n1000v# show ssh key	(Optional) Displays the SSH server keys.
Step 6	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

```
Example:
n1000v# config t
n1000v(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
n1000v(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

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

Configuring an OpenSSH Key

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already generated an SSH public key in OpenSSH format.
- The user account already exists.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**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 following command: **username** *name* **password** *pwd*

Example:

```
n1000v(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==
```

Step 3	**exit** **Example:** n1000v(config)# exit n1000v#	Exits Global Configuration mode and returns you to EXEC mode.
Step 4	**show user-account**	(Optional) Displays the user account configuration.

Example:

```
n1000v# 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==
```

Step 5	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring IETF or PEM Keys

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already generated an SSH public key in one of the following formats:
  - IETF SECSH format
  - Public Key Certificate in PEM format

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**copy** *server-file* **bootflash:***filename*	Downloads the file containing the SSH key from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP.

	Example:
	`n1000v# 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`
	`n1000v#`	

Step 2	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you in the CLI Global Configuration mode.

Step 3	**username** *username* **sshkey file bootflash:***filename* **Example:** `n1000v(config)# username User1 sshkey file bootflash:secsh_file.pub`	Configures the SSH public key.

Step 4	**exit**
	
	Example:
	`n1000v(config)# exit`
	`n1000v#`
		Exits Global Configuration mode and returns you to EXEC mode.

	Command	Purpose
Step 5	**show user-account**	(Optional) Displays the user account configuration.

Example:
```
n1000v# 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+V0cm7CC
LUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6
mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+f
FzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJN
U1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==
```

Step 6	**copy running-config startup-config** **Example:** `n1000v# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Starting SSH Sessions

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already obtained the hostname and, if needed, the username, for the remote device.
- The SSH server is already enabled on the remote device.

SUMMARY STEPS

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

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

DETAILED STEPS

	Command	Purpose

Step 1	**ssh** [root@]{*ip address* | *hostname*} [**vrf** *vrf-name*]
	
	Example:
	`n1000v(config)# 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`
		Creates an SSH IP session to a remote device using IP. The default VRF is the default VRF.

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.

BSUMMARY STEPS

1. **clear ssh hosts**

DETAILED STEPS

	Command	Purpose

Step 1	`clear ssh hosts`
	
	Example:
	`n1000v# clear ssh hosts`
		Clears the SSH host sessions.

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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- If you disable SSH, to enable it again you must first generate an SSH server key.

  See the "Generating SSH Server Keys" procedure on page 7-3.

SUMMARY STEPS

1. **config t**
2. **no feature ssh**
3. **show ssh server**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**no feature ssh** **Example:** n1000v(config)# no feature ssh XML interface to system may become unavailable since ssh is disabled n1000v(config)#	Disables the SSH server. The default is enabled.
Step 3	**show ssh server** **Example:** n1000v(config)# show ssh server ssh is not enabled n1000v(config)#	(Optional) Displays the SSH server configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- If you disable SSH, to enable it again you must first generate an SSH server key.

  See the "Generating SSH Server Keys" procedure on page 7-3.

SUMMARY STEPS

1. **config t**
2. **no feature ssh**
3. **no ssh key** [**dsa** | **rsa**]
4. **show ssh key**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you in the CLI Global Configuration mode.
Step 2	**no feature ssh** **Example:** n1000v(config)# no feature ssh	Disables the SSH server.
Step 3	**no ssh key** [**dsa** | **rsa**] **Example:** n1000v(config)# no ssh key rsa	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	**show ssh key** **Example:** n1000v(config)# show ssh key	(Optional) Displays the SSH server key configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Example:

```
n1000v# config t
n1000v(config)# no feature ssh
n1000v(config)# no ssh key rsa
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
```

```
************************************
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
************************************
n1000v#
```

Clearing SSH Sessions

Use this procedure to clear SSH sessions from the device.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**show users** **Example:** n1000v# show users	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** n1000v# clear line 0	Clears a user SSH session.
Step 3	**show users** **Example:** n1000v# show users	Displays user session information.

```
Example:
n1000v# 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)*
n1000v# clear line 0
n1000v# 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

To display the SSH configuration information, use one of the following commands:

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.

```
Example:
n1000v# show ssh key rsa
**************************************
rsa Keys generated:Mon Jul 28 09:49:18 2008

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAGEAv0a4p6VulQMW4AMgoPfApB2KegF3QTojCzed51iVQnEkNglnM7A/oEIZAtlVLY
k/PEzt+ED7lPal/8pomaqjgRxHSeK2gw1cJKSDbcYH5na8uox1Hr50eK0q2+ZfvMqV

bitcount:768
fingerprint:
```

```
76:6c:a0:5c:79:a6:ae:3d:cb:27:a1:86:62:fa:09:df
**************************************
```

SSH Example Configuration

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

Step 1 Disable the SSH server.

```
n1000v# config t
n1000v(config)# no feature ssh
```

Step 2 Generate an SSH server key.

```
n1000v(config)# ssh key rsa
generating rsa key(1024 bits).....
.generated rsa key
```

Step 3 Enable the SSH server.

```
n1000v(config)# feature ssh
```

Step 4 Display the SSH server key.

```
n1000v(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2007

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

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

Step 5 Specify the SSH public key in OpenSSH format.

```
n1000v(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=
```

Step 6 Save the configuration.

```
n1000v(config)# copy running-config startup-config
```

Example:

```
n1000v# config t
n1000v(config)# no feature ssh
n1000v(config)# ssh key rsa
generating rsa key(1024 bits).....
n1000v(config)# feature ssh
n1000v(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
************************************

n1000v(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZ19G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=
n1000v(config)# copy running-config startup-config
[########################################] 100%
n1000v(config)#
```

Additional References

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

- Related Documents, page 7-15
- Standards, page 7-15

Related Documents

Related Topic	Document Title
CLI	*Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)*
Telnet	Chapter 8, “Configuring Telnet”

Standards

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

Feature History for SSH

This section provides the SSH release history.

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

CHAPTER 8

Configuring Telnet

This chapter describes how to configure Telnet and includes the following topics:

Information About the Telnet Server

The Telnet protocol enables you 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.

Prerequisites for Telnet

Telnet has the following prerequisites:

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

Guidelines and Limitations

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

Default Setting

The following table lists the default setting for Telnet.

Parameters	Default
Telnet server	Enabled.

Configuring Telnet

This section includes the following topics:

Enabling the Telnet Server

Use this procedure to enable the Telnet server. The Telnet server is enabled by default, but you can use this procedure to re-enable it if necessary.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, the Telnet server is enabled.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**feature telnet** **Example:** n1000v(config)# feature telnet n1000v(config)#	Enables the Telnet server.
Step 3	**show telnet server** **Example:** n1000v(config)# show telnet server telnet service enabled n1000v(config)#	(Optional) Displays the Telnet server configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Copies these changes made in the running configuration to the startup configuration.

Starting an IP Telnet Session to a Remote Device

Use this procedure to start a Telnet session to a remote device.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.
- You have verified that the Telnet server is enabled on the remote device.
- You have already obtained the hostname for the remote device and, if needed, the username on the remote device.
- You have already verified that the Telnet server is enabled. If not you have enabled it using the "Enabling the Telnet Server" procedure on page 8-2. By default, the Telnet server is enabled.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**telnet** {*ip address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** n1000v# telnet 10.10.1.1	Creates an IP Telnet session to the specified destination. **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.

Clearing Telnet Sessions

Use this procedure to clear Telnet sessions.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following.

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**show users** **Example:** n1000v# **show users**	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** n1000v# **clear line 1**	Clears a user Telnet session.
Step 3	**show users** **Example:** n1000v# **show users**	Displays user session information.

```
Example:
n1000v# 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)*
n1000v# clear line 1
n1000v# 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)*
n1000v#
```

Verifying the Telnet Configuration

To display the Telnet configuration information, use one of the following commands:

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.

Example:

```
n1000v# show running-config security all
version 4.0(1)
username admin password 5 $1$xMw2Q/1S$ZEWRvyAxAJAFV0weuSPvg1  role network-admin
username user2 password 5 $1$byNNnnSP$xfXVKjE5UEScvriwX3Kyj0  role network-operator
username user2 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLU
kBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mW
oM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+fFzTG
YAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1
JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==
telnet server enable

banner motd # User Access Verification #

ssh key rsa 1024 force
no ssh key dsa force
ssh server enable
```

Additional References

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

- Related Documents, page 8-5
- Standards, page 8-6

Related Documents

Related Topic	Document Title
SSH	Chapter 7, “Configuring SSH”
CLI	*Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(4)*

Standards

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

Feature History for Telnet

This section provides the Telnet release history.

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

CHAPTER 9

Configuring an IP ACL

This chapter describes how to configure IP access control lists (ACLs).

This chapter includes the following sections:

Information About ACLs

An ACL is an ordered set of rules for filtering traffic. When the device determines that an ACL applies to a packet, it tests the packet against the rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies a default rule. The device processes packets that are permitted and drops packets that are denied. For more information, see the "Implicit Rules" section on page 9-3.

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

This section includes the following topics:

ACL Types and Applications

When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.

The following types of port ACLs are supported for filtering Layer 2 traffic:

- IP ACLs—The device applies IPv4 ACLs only to IP traffic.
- MAC ACLs—The device applies MAC ACLs only to non-IP traffic.

Order of ACL Application

ACLs are applied in the following order:

1. Incoming Port ACL
2. Outgoing Port ACL

About Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module.

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 in order to match the rule.

This section describes some of the options that you can use when you configure a rule. For information about every option, see the applicable **permit** and **deny** commands in the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

This section includes the following topics:

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host. How you specify the source and destination depends on whether you are configuring IP or MAC ACLs. For information about specifying source and destination, see the applicable **permit** and **deny** commands in the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Protocols

IP and MAC ACLs let 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.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the Ethertype number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

In 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.

For a list of the protocols that each type of ACL supports by name, see the applicable **permit** and **deny** commands in the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Implicit Rules

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules.

All IP 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
```

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

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- 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)

For information about all filtering options available in rules, see the applicable **permit** and **deny** commands in the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

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:

  ```
  n1000v(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:

  ```
  n1000v(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, 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 in IPv4 and MAC ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

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

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

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules. For more information, see the "Implicit Rules" section on page 9-3.

Prerequisites for IP ACLs

IP ACLs have the following prerequisites:

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

Guidelines and Limitations

IP ACLs have the following configuration guidelines and limitations:

- In most cases, ACL processing for IP packets are processed on the I/O modules. Management interface traffic is always processed on the supervisor module, which is slower.
- ACLs are not supported in port channels.

Default Settings

Table 9-1 lists the default settings for IP ACL parameters.

Table 9-1 Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs (see the "Implicit Rules" section on page 9-3)

Configuring IP ACLs

This section includes the following topics:

Creating an IP ACL

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **[no] ip access-list** {*name* | **match-local-traffic**}
3. [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. **statistics per-entry**
5. **show ip access-lists** *name*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**[no] ip access-list** {*name* | **match-local-traffic**} **Example:** n1000v(config)# ip access-list acl-01 n1000v(config-acl)# **Example:** n1000v(config)# ip access-list match-local-traffic n1000v(config-acl)#	Creates the named IP ACL (up to 64 characters in length) and enters IP ACL configuration mode. The **match-local-traffic** option enables matching for locally-generated traffic. The **no** option removes the specified access list.

	Command	Purpose
Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `n1000v(config-acl)# permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.
Step 4	**statistics per-entry** **Example:** `n1000v(config-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	**show ip access-lists** *name* **Example:** `n1000v(config-acl)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 6	**copy running-config startup-config** **Example:** `n1000v(config-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing an IP ACL

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

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see the "Changing Sequence Numbers in an IP ACL" section on page 9-10.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **ip access-list** *name*
3. [*sequence-number*] {**permit | deny**} *protocol source destination*
4. **no** {*sequence-number* | {**permit | deny**} *protocol source destination*}
5. [**no**] **statistics per-entry**
6. **show ip access-list** *name*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**ip access-list** *name* **Example:** n1000v(config)# ip access-list acl-01 n1000v(config-acl)#	Places you into IP ACL configuration mode for the specified ACL.
Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** n1000v(config-acl)# 100 permit ip 192.168.2.0/24 any	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*
Step 4	**no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} **Example:** n1000v(config-acl)# no 80	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.
Step 5	[**no**] **statistics per-entry** **Example:** n1000v(config-acl)# statistics per-entry	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	**show ip access-lists** *name* **Example:** n1000v(config-acl)# show ip access-lists acl-01	(Optional) Displays the IP ACL configuration.
Step 7	**copy running-config startup-config** **Example:** n1000v(config-acl)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Removing an IP ACL

You can remove an IP ACL from the device.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make sure that you know whether the ACL is applied to an interface.
- Removing an ACL does not affect the configuration of the interfaces where applied. Instead, the device considers the removed ACL to be empty.

SUMMARY STEPS

1. **config t**
2. **[no] ip access-list** *name*
3. **show ip access-list** *name* **summary**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**no ip access-list** *name* **Example:** `n1000v(config)# no ip access-list acl-01`	Removes the IP ACL that you specified by name from the running configuration.
Step 3	**show ip access-list** *name* **summary** **Example:** `n1000v(config)# show ip access-lists acl-01 summary`	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **resequence ip access-list** *name starting-sequence-number increment*
3. **show ip access-lists** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**resequence ip access-list** *name starting-sequence-number increment* **Example:** `n1000v(config)# resequence access-list ip acl-01 100 10`	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	**show ip access-lists** *name* **Example:** `n1000v(config)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a Port ACL

Use this procedure to configure a port ACL by applying an IPv4 or ACL to a Layer 2 interface physical port.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You can apply one port ACL to an interface.
- Make 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. For more information, see the "Creating an IP ACL" section on page 9-6 or the "Changing an IP ACL" section on page 9-7.
- An IP ACL can also be configured in a port profile. For more information, see the "Adding an IP ACL to a Port Profile" procedure on page 9-12.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *port*
3. **ip port access-group** *access-list* **[in | out]**
4. **show running-config aclmgr**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface vethernet** *port*
	
	Example:
	`n1000v(config)# interface vethernet 40`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified vEthernet interface.

Step 3	**ip port access-group** *access-list* [**in** | **out**] **Example:** `n1000v(config-if)# ip port access-group acl-l2-marketing-group in`	Applies an inbound or outbound IPv4 ACL to the interface. You can apply one port ACL to an interface.
Step 4	**show running-config aclmgr** **Example:** `n1000v(config-if)# show running-config aclmgr`	(Optional) Displays the ACL configuration.

	Command	Purpose
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Adding an IP ACL to a Port Profile

You can use this procedure to add an IP ACL to a port profile:

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already created the IP ACL to add to this port profile using the "Creating an IP ACL" procedure on page 9-6; and you know its name.
- If using an existing port profile, you have already created it and you know its name.
- If creating a new port profile, you know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- For more information about port profiles, see the *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*;
- You know the name of the IP access control list that you want to configure for this port profile.
- You know the direction of packet flow for the access list.

SUMMARY STEPS

1. **config t**
2. **port-profile** [**type** {**ethernet** | **vethernet**}] *profile-name*
3. **ip port access-group** *name* {**in** | **out**}
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**port-profile** [**type** {**ethernet** | **vethernet**}] *name* **Example:** n1000v(config)# port-profile AccessProf n1000v(config-port-prof)#	Enters port profile configuration mode for the named port profile.
Step 3	**ip port access-group** *name* {**in** | **out**} **Example:** n1000v(config-port-prof)# ip port access-group allaccess4 out	Adds the named ACL to the port profile for either inbound or outbound traffic.
Step 4	**show port-profile name** *profile-name* **Example:** n1000v(config-port-prof)# show port-profile name AccessProf	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-port-prof)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Applying an IP ACL to the Management Interface

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make 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. For more information, see the "Creating an IP ACL" section on page 9-6 or the "Changing an IP ACL" section on page 9-7.

SUMMARY STEPS

1. **config t**
2. **interface mgmt0**
3. **[no] ip access-group** *access-list* **[in | out]**
4. **show ip access-lists** *access-list*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI global configuration mode.
Step 2	**interface mgmt0** **Example:** n1000v(config)# interface mgmt0 n1000v(config-if)#	Places you into interface configuration mode for the management interface.
Step 3	**[no] ip access-group** *access-list* [**in** | **out**] **Example:** n1000v(config-if)# ip access-group telnet in n1000v(config-if)#	Applies a specified inbound or outbound IPv4 ACL to the interface. The no option removes the specified configuration.
Step 4	**show ip access-lists** *access-list* **Example:** n1000v(config-if)# show ip access-lists telnet summary IP access list telnet statistics per-entry Total ACEs Configured:2 Configured on interfaces: mgmt0 - ingress (Router ACL) Active on interfaces: mgmt0 - ingress (Router ACL)	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Verifying IP ACL Configurations

To display IP ACL configuration information, use the following commands:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists [*name*]	Displays all IPv4 access control lists (ACLs) or a named IPv4 ACL.

Command	Purpose
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 ACL

Use 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, then the **show ip access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL.

Example Configurations for IP ACL

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL to vEthernet interface 40:

```
ip access-list acl-01
    permit ip 192.168.2.0/24 any
interface vethernet 40
ip port access-group acl-01 in
```

The following example shows how to enable access list matching for locally-generated traffic:

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

Additional References

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

- Related Documents, page 9-16
- Standards, page 9-16

Related Documents

Related Topic	Document Title
ACL concepts.	*Information About ACLs, page 9-1*
Configuring interfaces.	*Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1)SV1(4)*
Configuring port profiles.	*Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples for Cisco Nexus 1000V commands.	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for IP ACL

This section provides the IP ACL release history.

Feature Name	Releases	Feature Information
IP ACL for mgmt0 interface	4.2(1) SV1(4)	
IP ACL	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 10

Configuring a MAC ACL

This chapter describes how to configure MAC access control lists (ACLs), and includes the following sections:

Information About MAC ACLs

MAC ACLs are ACLs that filter traffic using information in the Layer 2 header of each packet.

Prerequisites for MAC ACLs

MAC ACLs have the following prerequisites:

- You are familiar with MAC addressing and non-IP protocols to configure MAC ACLs.
- You are familiar with the concepts in the "Information About ACLs" section on page 9-1.

Guidelines and Limitations

MAC ACLs have the following configuration guidelines and limitations:

- In most cases, ACL processing for IP packets are processed on the I/O modules. Management interface traffic is always processed on the supervisor module, which is slower.
- ACLs are not supported in port channels.

Default Settings

Table 10-1 lists MAC ACL defaults.

Table 10-1 Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default
ACL rules	Implicit rules apply to all ACLs (see the "Implicit Rules" section on page 9-3)

Configuring MAC ACLs

This section includes the following topics:

- Creating a MAC ACL, page 10-2
- Changing a MAC ACL, page 10-3
- Removing a MAC ACL, page 10-5
- Changing Sequence Numbers in a MAC ACL, page 10-6
- Applying a MAC ACL as a Port ACL, page 10-7
- Adding a MAC ACL to a Port Profile, page 10-8

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 know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have a name to assign to the ACL you are creating.
- If you want to also add the ACL to a port-profile, you must know or do the following:
  - If using an existing port profile, you have already created it using the *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*; and you know its name.
  - If creating a new port profile, you know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
  - You know the direction of packet flow for the access list.

SUMMARY STEPS

1. **config t**
2. **mac access-list** *name*
3. {**permit** | **deny**} *source destination protocol*

4. **statistics per-entry**
5. **show mac access-lists** *name*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**mac access-list** *name*
	
	Example:
	`n1000v(config)# mac access-list acl-mac-01`
	`n1000v(config-mac-acl)#`
		Creates the MAC ACL and enters ACL configuration mode.

Step 3	{**permit** | **deny**} *source destination protocol* **Example:** `n1000v(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any`	Creates a rule in the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.
Step 4	**statistics per-entry** **Example:** `n1000v(config-mac-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	**show mac access-lists** *name* **Example:** `n1000v(config-mac-acl)# show mac access-lists acl-mac-01`	(Optional) Displays the MAC ACL configuration for verification.
Step 6	**copy running-config startup-config** **Example:** `n1000v(config-mac-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing a MAC ACL

Use this procedure to change an existing MAC ACL, for example, to add or remove rules.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are 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.

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

SUMMARY STEPS

1. **config t**
2. **mac access-list** *name*
3. [*sequence-number*] {**permit | deny**} *source destination protocol*
4. **no** {*sequence-number* | {**permit | deny**} *source destination protocol*}
5. [**no**] **statistics per-entry**
6. **show mac access-lists** *name*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**mac access-list** *name* **Example:** n1000v(config)# mac access-list acl-mac-01 n1000v(config-mac-acl)#	Places you in ACL configuration mode for the ACL that you specify by name.
Step 3	[*sequence-number*] {**permit** | **deny**} *source destination protocol* **Example:** n1000v(config-mac-acl)# 100 permit mac 00c0.4f00.00 0000.00ff.ffff any	(Optional) Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.
Step 4	**no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*} **Example:** n1000v(config-mac-acl)# no 80	(Optional) Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.
Step 5	[**no**] **statistics per-entry** **Example:** n1000v(config-mac-acl)# statistics per-entry	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.

	Command	Purpose
Step 6	**show mac access-lists** *name* **Example:** n1000v(config-mac-acl)# show mac access-lists acl-mac-01	(Optional) Displays the MAC ACL configuration.
Step 7	**copy running-config startup-config** **Example:** n1000v(config-mac-acl)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Removing a MAC ACL

Use this procedure to remove a MAC ACL.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make sure that you know whether the ACL is applied to an interface.
- You can remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, removed ACLs are considered empty.
- To find the interfaces that a MAC ACL is configured on, use the **show mac access-lists** command with the **summary** keyword.

SUMMARY STEPS

1. **config t**
2. **no mac access-list** *name*
3. **show mac access-lists** *name* **summary**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**no mac access-list** *name* **Example:** n1000v(config)# no mac access-list acl-mac-01 n1000v(config)#	Removes the specified MAC ACL from the running configuration.

	Command	Purpose
Step 3	**show mac access-lists** *name* **summary** **Example:** `n1000v(config)# show mac access-lists acl-mac-01 summary`	(Optional) Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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. For more information, see the "Changing Sequence Numbers in a MAC ACL" section on page 10-6.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **resequence mac access-list** *name starting-sequence-number increment*
3. **show mac access-lists** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**resequence mac access-list** *name starting-sequence-number increment* **Example:** n1000v(config)# resequence mac access-list acl-mac-01 100 10	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	**show mac access-lists** *name* **Example:** n1000v(config)# show mac access-lists acl-mac-01	(Optional) Displays the MAC ACL configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Applying a MAC ACL as a Port ACL

Use this procedure to apply a MAC ACL as a port ACL.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make sure that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application. For more information about configuring MAC ACLs, see the "Configuring MAC ACLs" section on page 10-2.
- A MAC ACL can also be applied to a port using a port profile. For information about port profiles, see the *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *port*
3. **mac port access-group** *access-list* [**in** | **out**]
4. **show running-config aclmgr**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**interface vethernet** *port* **Example:** n1000v(config)# interface vethernet 35 n1000v(config-if)#	Places you into Interface Configuration mode for the specified interface.
Step 3	**mac port access-group** *access-list* [**in** | **out**] **Example:** n1000v(config-if)# mac port access-group acl-01 in	Applies a MAC ACL to the interface.
Step 4	**show running-config aclmgr** **Example:** n1000v(config-if)# show running-config aclmgr	(Optional) Displays ACL configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Adding a MAC ACL to a Port Profile

You can use this procedure to add a MAC ACL to a port profile:

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already created the MAC ACL to add to this port profile using the "Creating a MAC ACL" procedure on page 10-2; and you know its name.
- If using an existing port profile, you have already created it and you know its name.
- If creating a new port profile, you know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- For more information about port profiles, see the *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*;
- You know the direction of packet flow for the access list.

SUMMARY STEPS

1. **config t**

2. **port-profile** [**type** {**ethernet** | **vethernet**}] *profile-name*
3. **mac port access-group** *name* {**in** | **out**}
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-profile** [**type** {**ethernet** | **vethernet**}] *name*
	
	Example:
	`n1000v(config)# port-profile AccessProf`
	`n1000v(config-port-prof)#`
		Enters port profile configuration mode for the named port profile.

Step 3	**mac port access-group** *name* {**in** | **out**} **Example:** `n1000v(config-port-prof)# mac port access-group allaccess4 out`	Adds the named ACL to the port profile for either inbound or outbound traffic.
Step 4	**show port-profile name** *profile-name* **Example:** `n1000v(config-port-prof)# show port-profile name AccessProf`	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-port-prof)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying MAC ACL Configurations

You can use the following commands to verify the MAC ACL configuration:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration. See Example 10-1 on page 10-10.
show running-config aclmgr	Displays the ACL configuration, including MAC ACLs and the interfaces they are applied to. See Example 10-2 on page 10-10.
show running-config interface	Displays the configuration of the interface to which you applied the ACL. See Example 10-3 on page 10-10.

Example 10-1 show mac access-list

```
n1000v# show mac access-list

MAC access list acl-mac-01
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
n1000v#
```

Example 10-2 show running-config aclmgr

```
n1000v# show running-config aclmgr

!Command: show running-config aclmgr
!Time: Mon Jan  3 15:53:50 2011

version 4.2(1)SV1(4)
mac access-list acl-mac-01
  10 permit 00c0.4f00.0000 0000.00ff.ffff any

interface Vethernet35
  mac port access-group acl-mac-01 in

n1000v#
```

Example 10-3 show running-config interface

```
n1000v# show running-config interface

!Command: show running-config interface
!Time: Mon Jan  3 15:58:25 2011

version 4.2(1)SV1(4)

interface mgmt0
  ip address 172.23.180.75/24

interface Vethernet35
  mac port access-group acl-mac-01 in

interface Vethernet1998

interface control0
  ip address 10.2.10.10/24

n1000v#
```

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.

Example Configurations for MAC ACLs

This example shows how to create MAC ACL acl-mac-01 to permit MAC 00c0.4f00.00.000.00ff.ffff for any protocol, and apply the ACL as a port ACL for outbound traffic on vEthernet interface 35.

```
config t
mac access-list acl-mac-01
  permit 00c0.4f00.0000 0000.00ff.ffff any
interface vethernet 35
mac port access-group acl-mac-01 out
```

This example shows how to add the MAC ACL *allaccess4* to the port profile *AccessProf*:

```
config t
port-profile AccessProf
mac port access-group allaccess4 out
show port-profile name AccessProf
port-profile AccessProf
  description: allaccess4
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    mac port access-group allaccess4 out
  evaluated config attributes:
    mac port access-group allaccess4 out
  assigned interfaces:
```

Additional References

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

- Related Documents, page 10-12
- Standards, page 10-12

Related Documents

Related Topic	Document Title
ACL concepts.	*Information About ACLs, page 9-1*
Configuring interfaces.	*Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1)SV1(4)*
Configuring port profiles.	*Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(4)*
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples for all Cisco Nexus 1000V commands.	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for MAC ACL

This section provides the MAC ACL release history.

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

CHAPTER 11

Configuring Port Security

This chapter describes how to configure port security and includes the following sections:

Information About Port Security

Port security lets you configure Layer 2 interfaces permitting 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.

This section includes the following topics:

Secure MAC Address Learning

The process of securing a MAC address is called learning. The number of addresses that can be learned is restricted, as described in the "Secure MAC Address Maximums" section on page 11-3. Address learning can be accomplished using the following methods on any interface where port security is enabled:

- Dynamic Method, page 11-2 (the default method)
- Sticky Method, page 11-2

Static Method

The static learning method lets you manually add or remove secure MAC addresses in the configuration of an interface.

A static secure MAC address entry remains in the configuration of an interface until you explicitly remove it. For more information, see the "Removing a Static or a Sticky Secure MAC Address from an Interface" section on page 11-10.

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.

Dynamic addresses are aged and dropped once the age limit is reached, as described in the "Dynamic Address Aging" section on page 11-2.

Dynamic addresses do not persist through restarts.

To remove a specific address learned by the dynamic method or to remove all addresses learned by the dynamic method on a specific interface, see the "Removing a Dynamic Secure MAC Address" section on page 11-11.

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 copying the running-configuration to the startup-configuration, **copy run start**.

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.

To remove a specific address learned by the sticky method, see the "Removing a Static or a Sticky Secure MAC Address from an Interface" section on page 11-10.

Dynamic Address Aging

MAC addresses 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 address age:

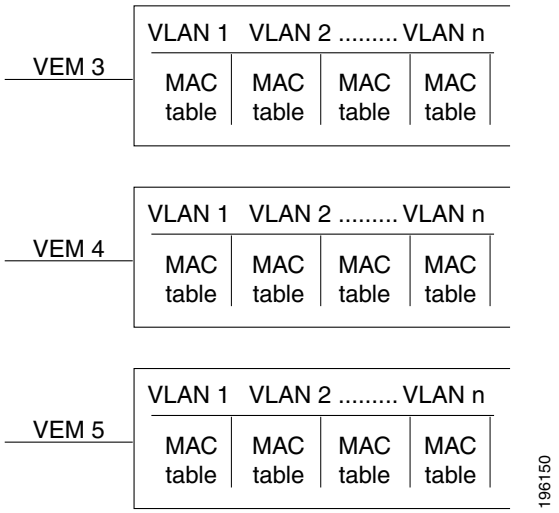
- Inactivity—The length of time after the device last received a packet from the address on the applicable interface.
- Absolute—The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Secure MAC Address Maximums

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

Figure 11-1 shows that each VLAN in a VEM has a forwarding table that can store a maximum number of secure MAC addresses. For current MAC address maximums, see Security Configuration Limits, page 16-1.

Figure 11-1 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.

Tip To make use of the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

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

- Device maximum—The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

- Interface maximum—You can configure a maximum number of secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Interface maximums cannot exceed the device maximum.
- VLAN maximum—You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. A VLAN maximum cannot exceed the interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

  For an example of how VLAN and interface maximums interact, see the "Security Violations and Actions" section on page 11-4.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first. To remove dynamically learned addresses, see the "Removing a Dynamic Secure MAC Address" section on page 11-11. To remove addresses learned by the sticky or static methods, see the "Removing a Static or a Sticky Secure MAC Address from an Interface" section on page 11-10.

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 5 addresses
  - The interface has a maximum of 10 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.

Example:

```
n1000v(config)# errdisable recovery cause psecure-violation
n1000v(config)# copy running-config startup-config (Optional)
```

- Protect—Prevents violations from occurring. Address learning continues until the maximum number of MAC addresses on the interface is reached, after which the device disables learning on the interface and drops all ingress traffic from nonsecure MAC addresses.

If a violation occurs because ingress traffic from a secure MAC address arrives at a different interface than the interface on which the address is secure, the action is applied on the interface that received the traffic. A MAC Move Violation is triggered on the port seeing the MAC which is already secured on another interface.

Port Security and Port Types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

- Access ports—You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN.
- Trunk ports—You can configure port security on interfaces that you have configured as Layer 2 trunk ports. VLAN maximums are not useful for access ports. The device allows VLAN maximums only for VLANs associated with the trunk port.
- SPAN ports—You can configure port security on SPAN source ports but not on SPAN destination ports.
- Ethernet 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 an access port to a trunk port on a Layer 2 interface configured with port security, all secure addresses learned by the dynamic method are dropped. The device to the native trunk VLAN moves the addresses learned by the static or sticky method.

Result of Changing a Trunk Port to an Access Port

When you change a trunk port to an access port on a Layer 2 interface configured with port security, all secure addresses learned by the dynamic method are dropped. All configured and sticky MAC addresses are dropped if they are not on the native trunk VLAN and do not match the access VLAN configured for the access port they are moving to.

Guidelines and Limitations

When configuring port security, follow these guidelines:

- Port security is not supported on the following:
  - Ethernet interfaces
  - Ethernet port-channel interfaces

  - Switched port analyzer (SPAN) destination ports
- Port security does not depend upon other features.
- Port security does not support 802.1X.
- Port Security cannot be configured on interfaces with existing static MACs.
- Port Security cannot be enabled on interfaces whose VLANs have an existing static MAC even if it is programmed on a different interface.

Default Settings

Table 11-1 lists the default settings for port security parameters.

Table 11-1 Default Port Security Parameters

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

Configuring Port Security

This section includes the following topics:

Enabling or Disabling Port Security on a Layer 2 Interface

Use this procedure to enable or disable port security on a Layer 2 interface. For more information about dynamic learning of MAC addresses, see the "Secure MAC Address Learning" section on page 11-1.

Note You cannot enable port security on a routed interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, port security is disabled on all interfaces.
- Enabling port security on an interface also enables dynamic MAC address learning. If you want to enable sticky MAC address learning, you must also complete the steps in the "Enabling or Disabling Sticky MAC Address Learning" section on page 11-8.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*
3. **[no] switchport port-security**
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

Step 3	**[no] switchport port-security** **Example:** `n1000v(config-if)# switchport port-security`	Enables port security on the interface. Using the **no** option disables port security on the interface.
Step 4	**show running-config port-security** **Example:** `n1000v(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling or Disabling Sticky MAC Address Learning

Use this procedure to disable or enable sticky MAC address learning on an interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Dynamic MAC address learning is the default on an interface.
- By default, sticky MAC address learning is disabled.
- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*
3. [**no**] **switchport port-security mac-address sticky**
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

	Command	Purpose

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

Step 3	**[no] switchport port-security mac-address sticky** **Example:** `n1000v(config-if)# 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	**show running-config port-security** **Example:** `n1000v(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Adding a Static Secure MAC Address on an Interface

Use this procedure to add a static secure MAC address on a Layer 2 interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, no static secure MAC addresses are configured on an interface.
- Determine if the interface maximum has been reached for secure MAC addresses (use the **show port-security** command).
- If needed, you can remove a secure MAC address. See one of the following:
  - "Removing a Static or a Sticky Secure MAC Address from an Interface" section on page 11-10
  - "Removing a Dynamic Secure MAC Address" section on page 11-11)
  - "Configuring a Maximum Number of MAC Addresses" section on page 11-12).
- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*

3. [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**interface** *type number* **Example:** n1000v(config)# interface vethernet 36 n1000v(config-if)#	Places you into Interface Configuration mode for the specified interface.
Step 3	[**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*] **Example:** n1000v(config-if)# switchport port-security mac-address 0019.D2D0.00AE	Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.
Step 4	**show running-config port-security** **Example:** n1000v(config-if)# show running-config port-security	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Removing a Static or a Sticky Secure MAC Address 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, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*
3. **no switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

Step 3	**no switchport port-security mac-address** *address* **Example:** `n1000v(config-if)# no switchport port-security mac-address 0019.D2D0.00AE`	Removes the MAC address from port security on the current interface.
Step 4	**show running-config port-security** **Example:** `n1000v(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a Dynamic Secure MAC Address

Use this procedure to remove a dynamically learned, secure MAC address.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **clear port-security dynamic** {**interface vethernet** *number* | **address** *address*} [**vlan** *vlan-ID*]
3. **show port-security address**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**clear port-security dynamic** {**interface vethernet** *number* | **address** *address*} [**vlan** *vlan-ID*] **Example:** n1000v(config)# clear port-security dynamic interface vethernet 36	Removes dynamically learned, secure MAC addresses, as specified. If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify. If you use the **address** keyword, you remove the single, dynamically learned address that you specify. Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.
Step 3	**show port-security address** **Example:** n1000v(config)# show port-security address	Displays secure MAC addresses.

Configuring a Maximum Number of MAC Addresses

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

Note When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the command is rejected.

To reduce the number of addresses learned by the sticky or static methods, see the "Removing a Static or a Sticky Secure MAC Address from an Interface" section on page 11-10.

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 know or do the following:

- You are logged in to the CLI in EXEC mode.
- The Secure MACs share the L2 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.
- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** ***type*** *number*
3. [**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

Step 3	[**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*] **Example:** `n1000v(config-if)# switchport port-security maximum 425`	Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 4096. The **no** option resets the maximum number of MAC addresses to the default, which is 1. If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.

	Command	Purpose
Step 4	**show running-config port-security** **Example:** `n1000v(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring an Address Aging Type and Time

Use this procedure to 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.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, the aging time is 0 minutes, which disables aging.
- Absolute aging is the default aging type.
- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** ***type*** *number*
3. [**no**] **switchport port-security aging type** {**absolute** | **inactivity**}
4. [**no**] **switchport port-security aging time** *minutes*
5. **show running-config port-security**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Places you into CLI Global Configuration mode.
Step 2	**interface** *type number* **Example:** n1000v(config)# interface vethernet 36 n1000v(config-if)#	Places you into Interface Configuration mode for the specified interface.
Step 3	[**no**] **switchport port-security aging type** {**absolute** | **inactivity**} **Example:** n1000v(config-if)# switchport port-security aging type inactivity	Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging.
Step 4	[**no**] **switchport port-security aging time** *minutes* **Example:** n1000v(config-if)# switchport port-security aging time 120	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	**show running-config port-security** **Example:** n1000v(config-if)# show running-config port-security	Displays the port security configuration.
Step 6	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring a Security Violation Action

Use this procedure to configure how an interface responds to a security violation.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The default security action is to shut down the port on which the security violation occurs.
- 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.

  For more information, see the "Security Violations and Actions" section on page 11-4.

- Make sure that port security is enabled on the interface that you are configuring.
  - To verify the configuration, see the "Verifying the Port Security Configuration" section on page 11-18.
  - To enable port security on the interface, see the "Enabling or Disabling Port Security on a Layer 2 Interface" section on page 11-6.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*
3. **[no] switchport port-security violation {protect | restrict | shutdown}**
4. **show running-config port-security**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

Step 3	**[no] switchport port-security violation {protect | restrict | shutdown}** **Example:** `n1000v(config-if)# switchport port-security violation protect`	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. • **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 increments the SecurityViolation counter. • **shutdown**: (the default) Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

	Command	Purpose
Step 4	**show running-config port-security** **Example:** `n1000v(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Recovering Ports Disabled for Port Security Violations

Use this procedure to automatically recover an interface disabled for port security violations.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- To recover an interface manually from the error-disabled state, you must enter the **shutdown** command and then the **no shutdown** command .
- For more information, see the "Security Violations and Actions" section on page 11-4.

SUMMARY STEPS

1. **config t**
2. **interface** *type number*
3. **errdisable recovery cause psecure-violation**
4. **errdisable recovery interval** *seconds*
5. **show interface** *type number*

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Places you into CLI Global Configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface vethernet 36`
	`n1000v(config-if)#`
		Places you into Interface Configuration mode for the specified interface.

	Command	Purpose
Step 3	**errdisable recovery cause psecure-violation** **Example:** `n1000v(config-if)# errdisable recovery cause psecure-violation`	Enables timed automatic recovery of the specified port that is disabled for port security violation.
Step 4	**errdisable recovery interval** *seconds* **Example:** `n1000v(config-if)# errdisable recovery interval 30`	Configures a timer recovery interval in seconds from 30 to 65535 seconds.
Step 5	**show interface** *type number* **Example:** `n1000v(config-if)# show running-config port-security`	Displays theinterface state for verification.

Verifying the Port Security Configuration

Use the following commands to display the port security configuration information:

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

For detailed information about the fields in the output from this command, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses. For detailed information about the fields in the output from this command, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Example Configuration for Port Security

The following example shows a port security configuration for vEthernet 36 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Protect.

```
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
```

Additional References

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

- Related Documents, page 11-19
- Standards, page 11-19

Related Documents

Related Topic	Document Title
Layer 2 switching	*Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(4)*
Port security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for Port Security

This section provides the port security feature release history.

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

CHAPTER 12

Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping and includes the following sections:

Information About DHCP Snooping

This section includes the following topics:

Overview

DHCP snooping acts 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. For more information about these features, see Chapter 13, "Configuring Dynamic ARP Inspection" and Chapter 14, "Configuring IP Source Guard."

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.

Trusted and Untrusted Sources

DHCP snooping identifies ports as trusted or untrusted. When you enable DHCP snooping, by default all vEthernet ports are untrusted and all ethernet ports (uplinks), port channels, special vEthernet ports (used by other features, such as VSD, for their operation) are trusted.You can configure whether DHCP trusts traffic sources.

In an enterprise network, a trusted source is a device that is under your administrative control. Any device beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Cisco Nexus 1000V, 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. This restriction prevents the uplink from being shut down for not conforming to rate limits or DHCP responses.

You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network or if the administrator is running the DHCP server in a VM. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the device through trusted interfaces.

DHCP Snooping Binding Database

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

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

DHCP snooping updates the database when the device receives specific DHCP messages. For example, the feature adds an entry to the database when the device receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the device receives a DHCPRELEASE 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.

You can remove dynamically added entries from the binding database by using the **clear ip dhcp snooping binding** command. For more information, see the "Clearing the DHCP Snooping Binding Database" section on page 12-13.

Relay Agent Information Option

You can configure DHCP to add the VSM MAC address and vEthernet port in the DHCP packet. This is called the DHCP Relay Agent Information Option, or Option 82, and is inserted by the DHCP relay agent when forwarding DHCP packets. Server administrators may use the information to implement IP address assignment policies.

The relay agent identifies the following:

Information Option	Description
circuit ID	vEthernet port name
remote ID	VSM MAC address

For detailed information about the Relay Agent Information Option, see *RFC-3046, DHCP Relay Agent Information Option.*

To configure the relay agent, see the "Relaying Switch and Circuit Information in DHCP" procedure on page 12-15.

High Availability

The DHCP snooping binding table and all database entries created on the VEM are exported to the VSM and are persistent across VSM reboots.

Prerequisites for DHCP Snooping

DHCP snooping has the following prerequisites:

- You must be familiar with DHCP to configure DHCP snooping.

Guidelines and Limitations

DHCP snooping has the following configuration guidelines and limitations:

- A DHCP snooping database is stored on each VEM and can contain up to 1024 bindings.
- 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 AIPC, management, and inband ports on a particular VEM), these virtual Ethernet interfaces must be configured as trusted interfaces.
- The connecting interfaces on a device upstream from the Cisco Nexus 1000V must be configured as trusted if DHCP snooping is enabled on the device.

Default Settings

Table 12-1 lists the defaults for DHCP snooping.

Table 12-1 Default DHCP Snooping Parameters

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.

Configuring DHCP Snooping

This section includes the following topics:

Minimum DHCP Snooping Configuration

The minimum configuration for DHCP snooping is as follows:

Step 1 Enable the DHCP feature. For more information, see the "Enabling or Disabling the DHCP Feature" section on page 12-5.

Step 2 Enable DHCP snooping globally. For more information, see the "Enabling or Disabling DHCP Snooping Globally" section on page 12-6.

Step 3 Enable DHCP snooping on at least one VLAN. For more information, see the "Enabling or Disabling DHCP Snooping on a VLAN" section on page 12-7.

By default, DHCP snooping is disabled on all VLANs.

Step 4 Ensure that the DHCP server is connected to the device using a trusted interface. For more information, see the "Configuring an Interface as Trusted or Untrusted" section on page 12-9.

Enabling or Disabling the DHCP Feature

Use this procedure to globally enable or disable the DHCP feature.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, DHCP is disabled.

SUMMARY STEPS

1. **config t**
2. **feature dhcp**
3. **show feature**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**feature dhcp**
	
	Example:
	`n1000v(config)# feature dhcp`
	
	Example:
	`n1000v(config)# no feature dhcp`
		Enables DHCP snooping globally. The **no** option disables DHCP snooping but preserves an existing DHCP snooping configuration.

	Command	Purpose

Step 3	**show feature**
	
	Example:
	`n1000v(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`
	`n1000v(config)#`
		Shows the state (enabled or disabled) of each available feature.

Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Enabling or Disabling DHCP Snooping Globally

Use this procedure to globally enable or disable the DHCP snooping.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- 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.

SUMMARY STEPS

1. **config t**
2. **[no] ip dhcp snooping**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**[no] ip dhcp snooping** **Example:** n1000v(config)# ip dhcp snooping	Enables DHCP snooping globally. The **no** option disables DHCP snooping but preserves an existing DHCP snooping configuration.
Step 3	**show running-config dhcp** **Example:** n1000v(config)# show running-config dhcp	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Enabling or Disabling DHCP Snooping on a VLAN

Use this procedure to enable or disable DHCP snooping on one or more VLANs.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, DHCP snooping is disabled on all VLANs.

SUMMARY STEPS

1. **config t**
2. **[no] ip dhcp snooping vlan** *vlan-list*
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:** n1000v(config)# ip dhcp snooping vlan 100,200,250-252	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP snooping on the VLANs specified.
Step 3	**show running-config dhcp** **Example:** n1000v(config)# show running-config dhcp	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Enabling or Disabling DHCP Snooping MAC Address Verification

Use this procedure to enable or disable DHCP snooping MAC address verification. If the device receives a packet on an untrusted interface and the source MAC address and the DHCP client hardware address do not match, address verification causes the device to drop the packet.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- MAC address verification is enabled by default.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping verify mac-address**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**[no] ip dhcp snooping verify mac-address** **Example:** n1000v(config)# ip dhcp snooping verify mac-address	Enables DHCP snooping MAC address verification. The **no** option disables MAC address verification.
Step 3	**show running-config dhcp** **Example:** n1000v(config)# show running-config dhcp	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring an Interface as Trusted or Untrusted

Use this procedure to configure whether a virtual interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust on the following:

- Layer 2 vEthernet interfaces
- Port Profiles for Layer 2 vEthernet interfaces

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, vEthernet interfaces are untrusted. The only exception is the special vEthernet ports used by other features such as VSD which are trusted
- Ensure that the vEthernet interface is configured as a Layer 2 interface.
- For seamless DHCP snooping, DAI, and 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.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*

   port-profile *profilename*
3. **[no] ip dhcp snooping trust**
4. **show running-config dhcp**

5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface vethernet** *interface-number*
	
	Example:
	`n1000v(config)# interface vethernet 3`
	`n1000v(config-if)#`
		Enters interface configuration mode, where *interface-number* is the vEthernet interface that you want to configure as trusted or untrusted for DHCP snooping.

	port-profile *profilename*
	
	Example:
	`n1000v(config)# port-profile vm-data`
	`n1000v(config-port-prof)#`
		Enters port profile configuration mode for the specified port profile, where *profilename* is a unique name of up to 80 characters.

Step 3	[**no**] **ip dhcp snooping trust** **Example:** `n1000v(config-if)# ip dhcp snooping trust`	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.
Step 4	**show running-config dhcp** **Example:** `n1000v(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

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 know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*

port-profile *profilename*

3. **[no] ip dhcp snooping limit rate** *rate*
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**interface vethernet** *interface-number* **Example:** n1000v(config)# interface vethernet 3 n1000v(config-if)#	Enters interface configuration mode, where *interface-number* is the vEthernet interface for which you want to configure the DHCP packets per second limit.
	port-profile *profilename* **Example:** n1000v(config)# port-profile vm-data n1000v(config-port-prof)#	Enters port profile configuration mode for the specified port profile, where *profilename* is a unique name of up to 80 characters.
Step 3	[**no**] **ip dhcp snooping limit rate** *rate* **Example:** n1000v(config-port-prof)# ip dhcp snooping limit rate 30	Configures the limit for the rate of DHCP packets per second (1 - 2048). The **no** option removes the rate limit.
Step 4	**show running-config dhcp** **Example:** n1000v(config-if)# show running-config dhcp	Shows the DHCP snooping configuration.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Detecting Ports Disabled for DHCP Rate Limit Violation

Use this procedure to globally configure detection of ports disabled for exceeding the DHCP rate limit.

BEFORE YOU BEGIN

Before beginning this procedures, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- A failure to conform to the set rate causes the port to be put into an errdisable state.
- You must enter the **shutdown** command and then the **no shutdown** command to recover an interface manually from the error-disabled state.

SUMMARY STEPS

1. **config t**
2. **[no] errdisable detect cause dhcp-rate-limit**
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**[no] errdisable detect cause dhcp-rate-limit** **Example:** `n1000v(config)# errdisable detect cause dhcp-rate-limit`	Enables DHCP error-disabled detection. The **no** option disables DHCP error-disabled detection.
Step 3	**show running-config dhcp** **Example:** `n1000v(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

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.

BEFORE YOU BEGIN

Before beginning this procedures, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Ports that rate causes the port to be put into an errdisable state.
- You must enter the **shutdown** command and then the **no shutdown** command to recover an interface manually from the error-disabled state.

SUMMARY STEPS

1. **config t**
2. **[no] errdisable recovery cause dhcp-rate-limit**
3. **errdisable recovery interval** *timer-interval*

4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**[no] errdisable recovery cause dhcp-rate-limit** **Example:** `n1000v(config)# errdisable detect cause dhcp-rate-limit`	Enables DHCP error-disabled recovery. The **no** option disables DHCP error-recovery.
Step 3	**errdisable recovery interval** *timer-interval* **Example:** `n1000v(config)# errdisable recovery interval 30`	Sets the DHCP error-disabled recovery interval, where *timer-interval* is the number of seconds (30-65535).
Step 4	**show running-config dhcp** **Example:** `n1000v(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Clearing the DHCP Snooping Binding Database

This section includes the following procedures:

- Clearing All Binding Entries, page 12-13
- Clearing Binding Entries for an Interface, page 12-14

Clearing All Binding Entries

Use this procedure to remove all entries from the DHCP snooping binding database.

BEFORE YOU BEGIN

Before beginning this procedures, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **clear ip dhcp snooping binding**

2. **show ip dhcp snooping binding**

DETAILED STEPS

	Command	Purpose
Step 1	**clear ip dhcp snooping binding** **Example:** `n1000v# clear ip dhcp snooping binding`	Clears dynamically added entries from the DHCP snooping binding database.
Step 2	**show ip dhcp snooping binding** **Example:** `n1000v# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Clearing Binding Entries for an Interface

Use this procedure to remove binding entries for an interface from the DHCP snooping database.

BEFORE YOU BEGIN

Before beginning this procedures, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have the following information for the interface:
  - VLAN ID
  - IP address
  - MAC address

SUMMARY STEPS

1. **clear ip dhcp snooping binding** [{**vlan** *vlan-id* **mac** *mac-addr* **ip** *ip-addr* **interface** *interface-id*} | **vlan** *vlan-id1* | **interface** *interface-id1*]
2. **show ip dhcp snooping binding**

DETAILED STEPS

	Command	Purpose
Step 1	**clear ip dhcp snooping binding** [{**vlan** *vlan-id* **mac** *mac-addr* **ip** *ip-addr* **interface** *interface-id*} | **vlan** *vlan-id1* | **interface** *interface-id1*] **Example:** `n1000v# clear ip dhcp snooping binding vlan 10 mac EEEE.EEEE.EEEE ip 10.10.10.1 interface vethernet 1`	Clears dynamically added entries for an interface from the DHCP snooping binding database.
Step 2	**show ip dhcp snooping binding** **Example:** `n1000v# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Relaying Switch and Circuit Information in DHCP

Use this procedure to globally configure relaying of the VSM MAC address and vEthernet port information in DHCP packets. This is also called Option 82 and Relay Agent Information Option.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- For more information, see the following:
  - "Relay Agent Information Option" section on page 12-3
  - *RFC-3046, DHCP Relay Agent Information Option.*

SUMMARY STEPS

1. **config t**
2. **[no] ip dhcp snooping information option**
3. **show runing-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**[no] ip dhcp snooping information option** **Example:** n1000v(config)# ip dhcp snooping information option n1000v(config)#	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	**show running-config dhcp** **Example:** n1000v(config)# show running-config dhcp !Command: show running-config dhcp !Time: Fri Dec 17 11:30:22 2010 version 4.2(1)SV1(4) ip dhcp snooping information option service dhcp ip dhcp relay ip dhcp relay information option n1000v(config)#	(Optional) Displays the DHCP snooping configuration for verification.

	Command	Purpose
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying the DHCP Snooping Configuration

To verify the DHCP snooping configuration, use the following commands:

Command	Purpose
show running-config dhcp	Displays the DHCP snooping configuration
show ip dhcp snooping	Displays general information about DHCP snooping.
show ip dhcp snooping binding	Display the contents of the DHCP snooping binding table.
show feature	Displays the features available, such as DHCP, and whether they are enabled.

For detailed information about these commands, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Monitoring DHCP Snooping

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping statistics. For detailed information about this command, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Example Configuration for DHCP Snooping

This example shows how to enable DHCP snooping on two VLANs, with vEthernet interface 5 trusted because the DHCP server is connected to that interface:

```
feature dhcp

interface vethernet 5
ip dhcp snooping trust
ip dhcp snooping vlan 1, 50
```

Additional References

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

- Related Documents, page 12-17
- Standards, page 12-17

Related Documents

Related Topic	Document Title
IP Source Guard	*Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1)SV1(4)*, Chapter 14, "Configuring IP Source Guard"
Dynamic ARP Inspection	*Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1)SV1(4)*, Chapter 13, "Configuring Dynamic ARP Inspection"
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

Standards	Title
RFC-2131	*Dynamic Host Configuration Protocol* (http://tools.ietf.org/html/rfc2131)
RFC-3046	*DHCP Relay Agent Information Option* (http://tools.ietf.org/html/rfc3046)

Feature History for DHCP Snooping

Table 12-2 lists the release history for this feature.

Table 12-2 Feature History for DHCP Snooping

Feature Name	Releases	Feature Information
Relay Agent (Option 82)	4.2(1)SV1(4)	You can configure relaying of VSM MAC and port information in DHCP packets.
feature dhcp command	4.2(1)SV1(4)	Command added for enabling DHCP feature globally.
DHCP snooping	4.0(4)SV1(2)	This feature was introduced.

CHAPTER 13

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI).

This chapter includes the following sections:

Information About DAI

This section includes the following topics:

About ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

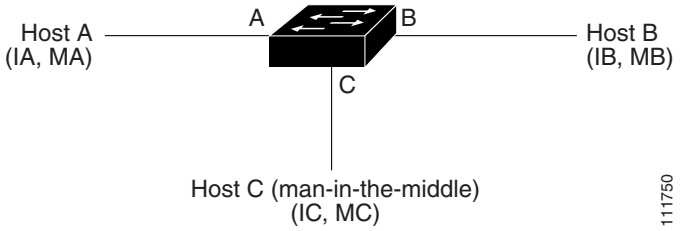
To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

About 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 their ARP caches. Figure 13-1 shows an example of ARP cache poisoning.

Figure 13-1 ARP Cache Poisoning



In Figure 13-1, 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 a host with an IP address of IA and a MAC address of MC
- one for a host with the IP address of IB and a MAC address of 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 and the device use MC as the destination MAC address for traffic intended for IB.

Because host C knows the authentic MAC addresses for IA and IB, it can forward the intercepted traffic.

About 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. For more information about trusted interfaces, see the Interface Trust and Network Security, page 13-3.

You can enable or disable validation of ARP packets for destination MAC address, source MAC address, and IP address. For more information, see the "Validating ARP Packets" section on page 13-13.

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

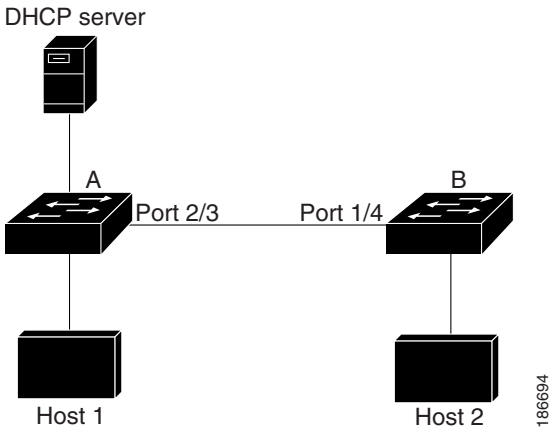
  Packets bypass all DAI validation checks.

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. For information about configuring a trusted interface, see the "Configuring a Trusted vEthernet Interface" section on page 13-6.

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In Figure 13-2, assume that both device A and device B are running DAI on the VLAN that includes host 1 and host 2. If host 1 and host 2 acquire their IP addresses from the DHCP server connected to device A, only device A binds the IP-to-MAC address of host 1. If the interface between device A and device B is untrusted, the ARP packets from host 1 are dropped by device B and connectivity between host 1 and host 2 is lost.

Figure 13-2 ARP Packet Validation on a VLAN Enabled for DAI



If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though device B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a device that runs DAI do not poison the ARP caches of other hosts in the network; however, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a device that runs DAI.

Note Depending on your network setup, you may not be able to validate a given ARP packet on all devices in the VLAN.

Prerequisites for DAI

The following are prerequisite to configuring DAI.

- You are familiar with the following:
  - ARP

    For more information, see IETF Standard RFC-826, *An Ethernet Address Resolution Protocol* (http://tools.ietf.org/html/rfc826).
  - DHCP Snooping

    For more information, see Configuring DHCP Snooping, page 12-1.
- The software running on your Cisco Nexus 1000V supports DAI.
- The VEM feature level is updated to a release that supports DAI.

  For more information about setting the VEM feature level, see the *Cisco Nexus 1000V Software Upgrade Guide, Release 4.2(1)SV1(4)*.

Guidelines and Limitations

DAI has the following configuration guidelines and limitations:

- 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 without DAI. This separation secures the ARP caches of hosts in the domain with DAI.
- DAI verifies IP-to-MAC address bindings in incoming ARP requests and ARP responses. If you have not configured static entries, then DHCP snooping must be enabled on the same VLANs on which you configure DAI. For more information, see the "Configuring DHCP Snooping" section on page 12-4.

- DAI is supported on vEthernet interfaces and private VLAN ports.
- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is configured. For more information, see the "Configuring DHCP Snooping" section on page 12-4).
- 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

Table 13-1 lists the DAI defaults.

Table 13-1 Default DAI Settings

Parameters	Default
VLAN	VLANs are not configured for DAI.
Trust state of vEthernet interfaces not in a VSD	Untrusted
Trust state of vEthernet Interfaces in a VSD	Trusted
Trust state of Ethernet port channels	Trusted
Incoming ARP packet rate limit for untrusted interfaces	15 packets per second (pps)
Incoming ARP packet rate limit for trusted interfaces	Unlimited
Rate limit burst interval	1 second
Detecting and Recovering DAI error-disabled interfaces	Error-disabled detection and recovery is not configured.
Validation checks	No checks are performed.
VLAN statistics	ARP request and response statistics.

Configuring DAI

This section includes the following topics:

Configuring a VLAN for DAI

Use this procedure to configure a VLAN or a list of VLANs for DAI.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, VLANs are not configured for DAI.
- You have already enabled DHCP snooping. For more information, see the "Enabling or Disabling the DHCP Feature" section on page 12-5.
- You know which VLANs you want to configure for DAI and they have already been created.

SUMMARY STEPS

1. **config t**
2. **[no] ip arp inspection vlan** *list*
3. **show ip arp inspection vlan** *list*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**ip arp inspection vlan** *list* **Example:** `switch(config)# ip arp inspection vlan 13`	Configures the specified VLAN or list of VLANs for DAI.
Step 3	**show ip arp inspection vlan** *list* **Example:** `switch(config)# show ip arp inspection vlan 13`	(Optional) Shows the DAI status for the specified list of VLANs.
Step 4	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring a Trusted vEthernet Interface

Use this procedure to configure a trusted vEthernet interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, vEthernet interfaces are untrusted, unless they are part of a VSD.
- If an interface is untrusted, all ARP requests and responses are verified for a valid IP-MAC address binding before the local cache is updated and the packet forwarded. If a packet has an invalid IP-MAC address binding, it is dropped.
- ARP packets 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.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*

   port-profile *profilename*
3. [**no**] **ip arp inspection trust**
4. **show ip arp inspection interface** *type slot/number*

   show port-profile *profilename*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**interface vethernet** *interface-number*
	Example:
	`switch(config)# interface vethernet 3`
	`switch(config-if)#`
		Places you into the CLI Interface Configuration mode, for the specified vEthernet interface.

	port-profile *profilename*
	Example:
	`switch(config)# port-profile vm-data`
	`switch(config-port-prof)#`
		Places you into the CLI Port Profile Configuration mode for the specified port profile.

Step 3	**ip arp inspection trust** **Example:** `switch(config-if)# ip arp inspection trust`	Configures the interface as a trusted ARP interface.

	Command	Purpose
	ip arp inspection trust **Example:** `switch(config-port-prof)# ip arp inspection trust`	Configures the interfaces assigned to the port profile as trusted ARP interfaces.
Step 4	**show ip arp inspection interface vethernet** *interface-number* **Example:** `switch(config-if)# show ip arp inspection interface vethernet 2`	(Optional) Displays the trusted state and the ARP packet rate for the specified interface.
	show port-profile *profilename* **Example:** `switch(config)# show port-profile vm-data`	(Optional) Displays the port profile configuration including the ARP trusted state.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Resetting a vEthernet Interface to Untrusted

Use this procedure to remove a trusted designation from a vEthernet interface, returning it to the default untrusted designation.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, vEthernet interfaces are untrusted, unless they are part of a VSD.
- If an interface is untrusted, all ARP requests and responses are verified for a valid IP-MAC address binding before the local cache is updated and the packet forwarded. If a packet has an invalid IP-MAC address binding, it is dropped.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*
3. **default ip arp inspection trust**
4. **show ip arp inspection interface** *type slot/number*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**interface vethernet** *interface-number*
	Example:
	`switch(config)# interface vethernet 3`
	`switch(config-if)#`
		Places you into the CLI Interface Configuration mode, for the specified vEthernet interface.

Step 3	**default ip arp inspection trust** **Example:** `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	**show ip arp inspection interface vethernet** *interface-number* **Example:** `switch(config-if)# show ip arp inspection interface vethernet 3`	(Optional) Displays the trusted state and the ARP packet rate for the specified interface.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring DAI Rate Limits

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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 = unlimited
  - Burst interval = 1 second
- 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.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*

   port-profile *profilename*
3. **ip arp inspection limit** {**rate** *pps* [**burst interval** *bint*] | **none**}
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**interface vethernet** *interface-number*
	Example:
	`switch(config)# interface vethernet 3`
	`switch(config-if)#`
		Places you into the CLI Interface Configuration mode, for the specified vEthernet interface.

	port-profile *profilename*
	Example:
	`switch(config)# port-profile vm-data`
	`switch(config-port-prof)#`
		Places you into the CLI Port Profile configuration mode for the specified port profile.

Step 3	**ip arp inspection limit** {**rate** *pps* [**burst interval** *bint*]	**none**}
		Example:
		`switch(config-if)# ip arp inspection limit rate 30`
		Example:
		`switch(config-port-prof)# ip arp inspection limit rate 30`
			Configures the specified ARP inspection limit on the interface or the port profile as follows:
			• **rate**: allowable values are between 1 and 2048 packets per second (pps)
			– Untrusted interface default = 15 packets per second
			– Trusted interface default = unlimited
			• **burst interval**: allowable values are between 1 and 15 seconds (default = 1 second).
			• **none**: unlimited number of packets per second

Step 4	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Resetting DAI Rate Limits to Default Values

Use this procedure to set the rate limit of ARP requests and responses to the defaults, removing any configured values.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The default DAI rate limits are as follows:
  - Untrusted interfaces = 15 packets per second
  - Trusted interfaces = unlimited
  - Burst interval = 1 second
- 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.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*
3. **default ip arp inspection limit** {**rate** *pps* [**burst interval** *bint*] | **none**}
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**interface vethernet** *interface-number*
	Example:
	`switch(config)# interface vethernet 3`
	`switch(config-if)#`
		Places you into the CLI Interface Configuration mode, for the specified vEthernet interface.

	Command	Purpose
Step 3	**default ip arp inspection limit** {**rate** *pps* [**burst interval** *bint*] | **none**} **Example:** `switch(config-if)# default ip arp inspection limit rate`	Removes the configured DAI rate limits from the interface and returns them to the default values. • **rate**: – Untrusted interface default = 15 packets per second – Trusted interface default = unlimited • **burst interval**: default = 1 second • **none**: unlimited number of packets per second
Step 4	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI rate limits.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Detecting and Recovering Error-Disabled Interfaces

Use this procedure to configure the detection and recovery of error-disabled interfaces.

BEFORE YOU BEGIN

Before beginning this procedures, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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**

SUMMARY STEPS

1. **config t**
2. **[no] errdisable detect cause arp-inspection**
3. **[no] errdisable recovery cause arp-inspection**
4. **errdisable recovery interval** *timer-interval*
5. **show running-config | include errdisable**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**errdisable detect cause arp-inspection** **Example:** `switch(config)# 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	**errdisable recovery cause arp-inspection** **Example:** `switch(config)# errdisable recovery cause arp-inspection`	Configures the recovery of interfaces that have been error-disabled by ARP inspection.
Step 4	**errdisable recovery interval** *timer-interval* **Example:** `switch(config)# errdisable recovery interval 30`	Configures the recovery interval for interfaces that have been error-disabled by ARP inspection. **timer-interval**: allowable values are between 30 and 65535 seconds.
Step 5	**show running-config | include errdisable** **Example:** `switch(config)# show running-config | include errdisable`	(Optional) Displays the errdisable configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Validating ARP Packets

Use this procedure to configure the validation of ARP packets.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- 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.

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

SUMMARY STEPS

1. **config t**
2. [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}
3. **show running-config dhcp**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you into the CLI Global Configuration mode.

Step 2	**ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]} **Example:** `switch(config)# ip arp inspection validate src-mac dst-mac ip`	Enables the specified validation and overwrites any existing validation that was previously saved: • Source MAC • Destination MAC • IP You can specify all three of these validations but you must specify at least one. Use the no option to disable a validation.
Step 3	**show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	(Optional) Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying the DAI Configuration

To display and verify the DAI configuration, use the following commands:

Command	Purpose
show running-config dhcp	Displays the DAI configuration.
show ip arp inspection	Displays the status of DAI.

Command	Purpose
show ip arp inspection interface vethernet *interface-number*	Displays the trust state and ARP packet rate for a specific interface.
show ip arp inspection vlan *vlan-ID*	Displays the DAI configuration for a specific VLAN.

For detailed information about command output, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Monitoring DAI

To monitor DAI, use the following commands:

Command	Purpose
show ip arp inspection statistics	Displays DAI statistics.
show ip arp inspection statistics vlan	Displays DAI statistics for a specified VLAN.
clear ip arp inspection statistics	Clears DAI statistics.

For detailed information about command output, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Example DAI Configuration

This example shows 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.

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. For configuration information, see Chapter 12, “Configuring DHCP Snooping.”

The following steps are used to configure DAI for this example:

Step 1 Enable DAI on VLAN 1 and verify the configuration.

```
n1000v# config t
n1000v(config)# ip arp inspection vlan 1
n1000v(config)# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
n1000v(config)#
```

Step 2 Check the statistics before and after DAI processes any packets.

```
n1000v# show ip arp inspection statistics vlan 1

Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
n1000v#
```

If VM 1 sends out two ARP requests with an IP address of 10.0.0.1 and a MAC address of 0002.0002.0002, both requests are permitted, as shown in the following command output:

```
n1000v# show ip arp inspection statistics vlan 1

Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
```

If 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])
```

The statistics display as follows:

```
n1000v# show ip arp inspection statistics vlan 1
n1000v#

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
n1000v#
```

Additional References

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

- Related Documents, page 13-17
- Standards, page 13-17

Related Documents

Related Topic	Document Title
DHCP snooping	Configuring DHCP Snooping, page 12-1
DAI and DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

Standards	Title
RFC-826	*An Ethernet Address Resolution Protocol* (http://tools.ietf.org/html/rfc826)

Feature History for DAI

Table 13-2 lists the release history for the DAI feature.

Table 13-2 Feature History for DAI

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

CHAPTER 14

Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on Cisco Nexus 1000Vs.

This chapter includes the following sections:

Information About IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches the IP and MAC address bindings of dynamic or static IP source entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table.

You can enable IP Source Guard on Layer 2 interfaces that are not trusted by DHCP snooping. IP Source Guard supports interfaces that are configured to operate in access mode and trunk mode. When you initially enable IP Source Guard, all inbound IP traffic on the interface is blocked except for the following:

- DHCP packets, which DHCP snooping inspects and then forwards or drops, depending upon the results of inspecting the packet.
- IP traffic from static IP source entries that you have configured in the 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.

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 forward the packet only if the MAC address of the packet is 00:02:B3:3F:3B:99.

Prerequisites for IP Source Guard

IP Source Guard has the following prerequisites:

- You should be familiar with DHCP snooping before you configure IP Source Guard.
- DHCP snooping is enabled (see the "Configuring DHCP Snooping" section on page 12-4).

Guidelines and Limitations

IP Source Guard has the following configuration guidelines and limitations:

- IP Source Guard limits IP traffic on an interface to only those sources that have an IP-MAC address binding table entry or static IP source entry. When you first enable IP Source Guard on an interface, you may experience disruption in IP traffic until the hosts on the interface receive a new IP address from a DHCP server.
- IP Source Guard is dependent upon DHCP snooping to build and maintain the IP-MAC address binding table or upon manual maintenance of static IP source entries. For more information on DHCP snooping, see Chapter 12, "Configuring DHCP Snooping."
- 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

Table 14-1 lists IP Source Guard defaults.

Table 14-1 Default IP Source Guard Parameters

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

Configuring IP Source Guard

This section includes the following topics:

Enabling or Disabling IP Source Guard on a Layer 2 Interface

Use this procedure to enable or disable IP Source Guard on a Layer 2 interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- By default, IP Source Guard is disabled on all interfaces.
- Ensure that DHCP snooping is enabled. For more information, see the "Enabling or Disabling the DHCP Feature" section on page 12-5.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*

   port-profile *profilename*
3. **[no] ip verify source dhcp-snooping-vlan**
4. **show running-config dhcp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	**interface vethernet** *interface-number* **Example:** switch(config)# interface vethernet 3 switch(config-if)#	Enters interface configuration mode, where *interface-number* is the vEthernet interface that you want to configure as trusted or untrusted for DHCP snooping.
	port-profile *profilename* **Example:** switch(config)# port-profile vm-data switch(config-port-prof)#	Enters port profile configuration mode for the specified port profile, where *profilename* is a unique name of up to 80 characters.
Step 3	**[no] ip verify source dhcp-snooping-vlan** **Example:** switch(config-if)# ip verify source dhcp-snooping vlan	Enables IP Source Guard on the interface. The **no** option disables IP Source Guard on the interface.
Step 4	**show running-config dhcp** **Example:** switch(config-if)# show running-config dhcp	(Optional) Displays the running configuration for DHCP snooping, including the IP Source Guard configuration.

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

Adding or Removing a Static IP Source Entry

Use this procedure to add or remove a static IP source entry on a device.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- By default, there are no static IP source entries on a device.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface vethernet** *interface-number*
3. **show ip dhcp snooping binding [interface vethernet** *interface-number*]
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters global configuration mode.
Step 2	[**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface vethernet** *interface-number* **Example:** switch(config)# ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface ethernet 3	Creates a static IP source entry for the current interface, or if you use the **no** option, removes a static IP source entry.
Step 3	**show ip dhcp snooping binding** [**interface vethernet** *interface-number*] **Example:** switch(config)# show ip dhcp snooping binding interface ethernet 3	(Optional) Displays IP-MAC address bindings for the interface specified, including static IP source entries. Static entries appear with the term “static” in the Type column.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Verifying the IP Source Guard Configuration

To display IP Source Guard configuration information, use one of the following commands:

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

For detailed information about command output, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*.

Displaying IP Source Guard Bindings

Use the **show ip verify source** command to display IP-MAC address bindings.

Example Configuration for IP Source Guard

The following example shows how to create a static IP source entry and then how to enable IP Source Guard on an interface:

```
ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface vethernet 3
interface ethernet 2/3
  no shutdown
  ip verify source dhcp-snooping-vlan
```

Additional References

For additional information related to implementing IP Source Guard, see the following sections:

- Related Documents, page 14-5
- Standards, page 14-6

Related Documents

Related Topic	Document Title
Information About DHCP Snooping, page 12-1	*Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1)SV1(4), Chapter 12, "Configuring DHCP Snooping"*
IP Source Guard commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for IP Source Guard

Table 14-2 lists the release history for this feature.

Table 14-2 Feature History for IP Source Guard

Feature Name	Releases	Feature Information
IP Source Guard	4.0(4)SV1(2)	This feature was introduced.

CHAPTER 15

Disabling HTTP Server

This chapter describes how to disable the HTTP server and includes the following topics:

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).

If you want to turn off the HTTP server, see the following "Guidelines and Limitations".

Guidelines and Limitations

- The HTTP server is enabled by default.
- VUM will not install VEMs if the HTTP server is disabled. During VEM installation, VUM talks directly to the HTTP server to extract required module information from the VSM. To install VEMs, you must do one of the following:
  - Use VUM by enabling the HTTP server during VEM installation, and then disabling it after the VEMs are installed.
  - Install VEMs manually without using VUM.
- The HTTP server must be enabled in order to get the Cisco Nexus 1000V XML plugin from the VSM.

Default Setting

The HTTP server is enabled by default.

Disabling HTTP Server

Use this procedure to disable the HTTP server.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- By default, the HTTP server is enabled.

SUMMARY STEPS

1. **config t**
2. **no feature http-server**
3. **show http-server**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters CLI global configuration mode.

Step 2	**no feature http-server**
	
	Example:
	`n1000v(config)# no feature http-server`
	`n1000v(config)#`
		Disables the HTTP server.

Step 3	**show http-server**
	
	Example:
	`n1000v(config)# show http-server`
	`http-server disabled`
		(Optional) Displays the HTTP server configuration (enabled or disabled).

Step 4	**copy running-config startup-config**
	
	Example:
	`n1000v(config)# copy running-config startup-config`
	`[########################################]`
	`100%`
	`n1000v(config)#`
		(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Example:

```
config t
no feature http-server
```

Verifying the HTTP Configuration

To display the HTTP configuration, use the following commands:

Command	Purpose
show http-server	Displays the HTTP server configuration. See Example 15-1
show feature	Displays the features available, such as LACP, and whether they are enabled. See Example 15-2

Example 15-1 show http-server

```
n1000v(config)# show http-server
http-server enabled
n1000v(config)#
```

Example 15-2 show feature

```
n1000v(config)# show feature
Feature Name          Instance  State
--------------------  --------  --------
dhcp-snooping         1         disabled
http-server           1         disabled
ippool                1         disabled
lacp                  1         disabled
netflow               1         disabled
private-vlan          1         disabled
sshServer             1         enabled
tacacs                1         disabled
telnetServer          1         disabled
n1000v(config)#
```

Additional References

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

- Related Documents, page 15-4
- Standards, page 15-4

Related Documents

Related Topic	Document Title
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)*

Standards

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

Feature History for Disabling the HTTP Server

This section provides the release history for disabling the HTTP server.

Feature Name	Releases	Feature Information
Disable HTTP server	4.2(1)SV1(4)	This feature was introduced.

CHAPTER 16

Security Configuration Limits

Table 16-1 shows the maximum configuration limits for Security features.

Table 16-1 Security Maximum Configuration Limits

Security Feature	Maximum Limit	
Active VLANs across all VEMs	2000	
MAC addresses over VLAN within a VEM	32000	
MAC addresses per VLAN within a VEM	4000	
Secure vEths per VSM	2000	
Secure MACs per VSM	8000	
Secure MACs per vEth	1025	
ACLs	128	
ACEs per ACL	128	
	Per DVS	**Per Host**
ACL Interfaces	2048	256
NetFlow Policies	32	8
NetFlow Interfaces	256	32
SPAN/ERSPAN Sessions	64	4
Port Security	2K	216
Multicast Groups	512	64
Virtual Service Domains (VSD)	64	6
VSD Interfaces	2048	214

INDEX

A

C

D

E

F

H

I

L

M

O

P

R

S

T

U

V