Cisco Nexus 3548 Switch NX-OS Security Configuration Guide, Release 5.0(3)A1(1)

First Published: November 05, 2012

Last Modified: December 15, 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-27858-01

CONTENTS

Preface

This preface contains the following sections:

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element(keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

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

This document uses the following conventions:

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

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

Related Documentation for Nexus 3548 Switch NX-OS Software

The Cisco Nexus 3548 Switch NX-OS documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps11541/tsd_products_support_series_home.html

Release Notes

The release notes are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/prod_release_notes_list.html

Installation and Upgrade Guides

The installation and upgrade guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/prod_installation_guides_list.html

License Information

For information about feature licenses in NX-OS, see the *Cisco NX-OS Licensing Guide*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/nx-os/licensing/guide/b_Cisco_NX-OS_Licensing_Guide.html.

For the NX-OS end user agreement and copyright information, see *License and Copyright Information for Cisco NX-OS Software*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.

Configuration Guides

The configuration guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html

Command References

The command references are available at the following URL:

http://cisco.com/en/US/products/ps11541/prod_command_reference_list.html

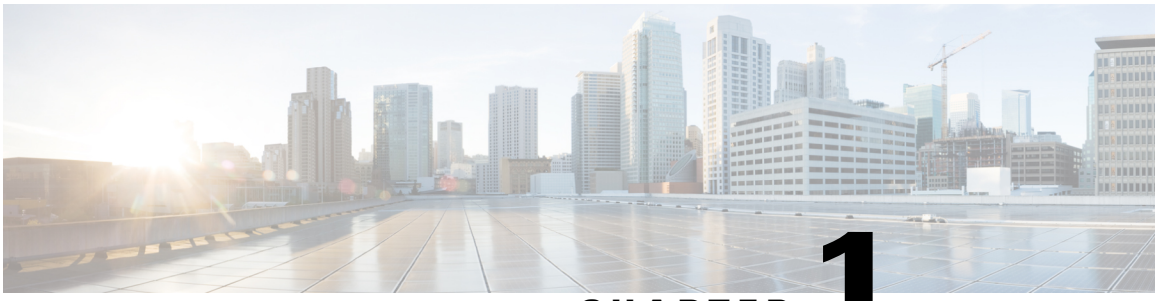
Error and System Messages

The error and system message reference guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/products_system_message_guides_list.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com . We appreciate your feedback.



CHAPTER 1

Overview

This chapter contains the following sections:

Authentication, Authorization, and Accounting

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

Authentication

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

Authorization

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

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

Accounting

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

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

RADIUS and TACACS+ Security Protocols

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

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

RADIUS

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

TACACS+

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

SSH and Telnet

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

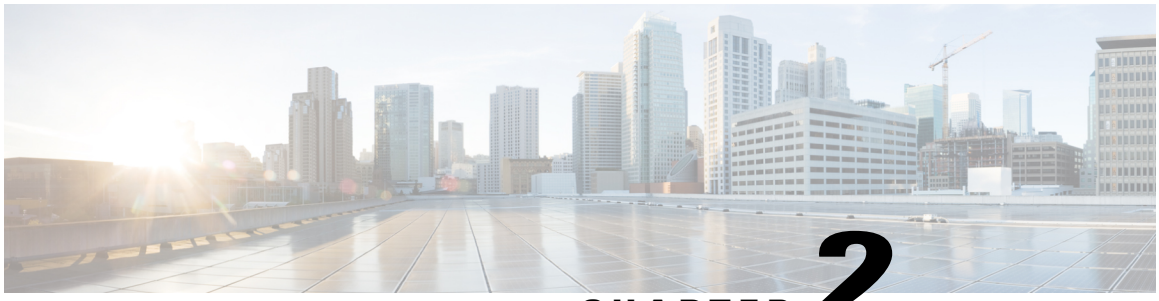
The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

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

IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When

the Cisco NX-OS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.



CHAPTER 2

Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

Information About AAA

AAA Security Services

The authentication, authorization, and accounting (AAA) features allows you to verify the identity of, grant access to, and track the actions of users who manage Cisco Nexus devices. The Cisco Nexus device supports Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control device Plus (TACACS+) protocols.

Based on the user ID and password that you provide, the switches perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the switch and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- Authentication—Identifies users, including login and password dialog, challenge and response, messaging support, and, encryption depending on the security protocol that you select.

- Authorization—Provides access control.

  Authorization to access a Cisco Nexus device is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- Accounting—Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

Note The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch in the fabric are easier to manage.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- The accounting log for all switches in the fabric can be centrally managed.
- User attributes for each switch in the fabric are easier to manage than using the local databases on the switches.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. A server group provides for failover servers if a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, that server group option is considered a failure. If required, you can specify multiple server groups. If a switch encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

On Cisco Nexus devices, you can have separate AAA configurations for the following services:

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

The following table lists the CLI commands for each AAA service configuration option.

Table 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**
User session accounting	**aaa accounting default**

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

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

Note If the method is for all RADIUS servers, instead of a specific server group, the Cisco Nexus devices choose the RADIUS server from the global pool of configured RADIUS servers in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco Nexus devices.

The following table describes the AAA authentication methods that you can configure for the AAA services.

Table 2: AAA Authentication Methods for AAA Services

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

Note For console login authentication, user login authentication, and user management session accounting, the Cisco Nexus devices try each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Logins

The authentication and authorization process for user login is as occurs:

- When you log in to the required Cisco Nexus device, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Cisco Nexus device sends an authentication request to the first AAA server in the group as follows:

  If the AAA server fails to respond, then the next AAA server is tried and so on until the remote server responds to the authentication request.

  If all AAA servers in the server group fail to respond, the servers in the next server group are tried.

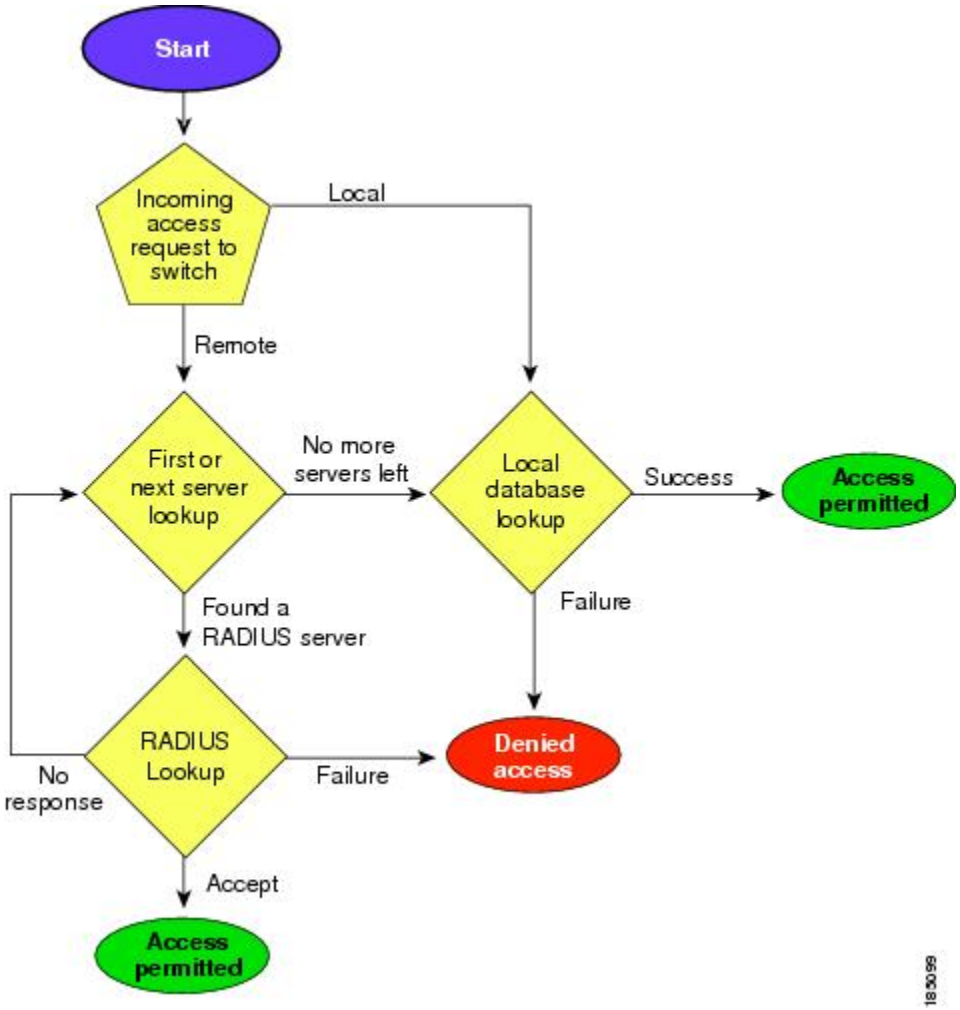
  If all configured methods fail, the local database is used for authentication.
- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

  If the AAA server protocol is RADIUS, user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.

  If the AAA server protocol is TACACS+, another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If your username and password are successfully authenticated locally, the Cisco Nexus device logs you in and assigns you the roles configured in the local database.

The following figure shows a flowchart of the authentication and authorization process.

Figure 1: Authentication and Authorization Flow for User Login



Note "No more server groups left" means that there is no response from any server in all server groups. "No more servers left" means that there is no response from any server within this server group.

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable.
- The Cisco Nexus device is configured as a client of the AAA servers.

- The preshared secret key is configured on the Cisco Nexus device and on the remote AAA servers.
- The remote server responds to AAA requests from the Cisco Nexus device.

Guidelines and Limitations for AAA

The Cisco Nexus devices do not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally. If an all numeric username exists on an AAA server and is entered during a login, the Cisco Nexus device still logs in the user.

Caution You should not create user accounts with usernames that are all numeric.

Configuring AAA

Configuring Console Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the Cisco Nexus device.
- Username only (**none**)

The default method is local.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command are used for a set of previously defined RADIUS servers. Use the **radius server-host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Before you configure console login authentication methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

	Command or Action	Purpose
		• **radius —**Uses the global pool of RADIUS servers for authentication. • *named-group* **—**Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show aaa authentication**	(Optional) Displays the configuration of the console login authentication methods.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure authentication methods for the console login:

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

Configuring Default Login Authentication Methods

The default method is local.

Before you configure default login authentication methods, configure RADIUS or TACACS+ server groups as needed. To configure default login authentication methods, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius —**Uses the global pool of RADIUS servers for authentication.

	Command or Action	Purpose
		• *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local** , which is used when no methods are configured or when all of the configured methods do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show aaa authentication**	(Optional) Displays the configuration of the default login authentication methods.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by the local user database if the remote AAA servers do not respond. If you have enabled the displaying of login failure messages, the following message is displayed:

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

Procedure

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

Configuring AAA Command Authorization

When a TACACS+ server authorization method is configured, you can authorize every command that a user executes with the TACACS+ server which includes all EXEC mode commands and all configuration mode commands.

The authorization methods include the following:

- Group—TACACS+ server group
- Local—Local role-based authorization
- None—No authorization is performed

The default method is Local.

Note There is no authorization on the console session.

Before You Begin

You must enable TACACS+ before configuring AAA command authorization.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authorization** {**commands** | **config-commands**} {**default**} {{[**group** *group-name*] | [**local**]} | {[**group** *group-name*] | [**none**]}}
	
	Example:
	`switch(config)# aaa authorization config-commands default group tac1`
	
	Example:
	`switch# aaa authorization commands default group tac1`
		Configures authorization parameters.
		Use the **commands** keyword to authorize EXEC mode commandes.
		Use the **config-commands** keyword to authorize configuration mode commands.
		Use the **group**, **local**, or **none** keywords to identify the authorization method.

The following example shows how to authorize EXEC mode commands with TACACS+ server group *tac1*:

```
switch# aaa authorization commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

```
switch(config)# aaa authorization config-commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, the command is authorized based on the user's *local* role.

```
switch(config)# aaa authorization config-commands default group tac1 local
```

The followng example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, allow the command regardless of the local role.

```
switch# aaa authorization commands default group tac1 none
```

The following example shows how to authorize EXEC mode commands regardless of the local role:

```
switch# aaa authorization commands default none
```

The following example shows how to authorize EXEC mode commands using the local role for authorization:

```
switch# aaa authorization commands default local
```

Enabling MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. You can use MSCHAP for user logins to a Cisco Nexus device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco Nexus device uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MSCHAP, you must configure your RADIUS server to recognize the MSCHAP vendor-specific attributes (VSAs).

The following table describes the RADIUS VSAs required for MSCHAP.

Table 3: MSCHAP RADIUS VSAs

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

Vendor-ID Number	Vendor-Type Number	VSA	Description
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP user in response to the challenge. It is only used in Access-Request packets.

Procedure

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

Configuring AAA Accounting Default Methods

The Cisco Nexus device supports TACACS+ and RADIUS methods for accounting. The switches report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco Nexus device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:.

- RADIUS server group—Uses the global pool of RADIUS servers for accounting.
- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.
- Local—Uses the local username or password database for accounting.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before You Begin

Before you configure AAA accounting default methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **aaa accounting default** {**group** *group-list* | **local**}	Configures the default accounting method. One or more server group names can be specified in a space-separated list. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius —**Uses the global pool of RADIUS servers for accounting. • *named-group* **—**Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is **local**, which is used when no server groups are configured or when all the configured server group do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show aaa accounting**	(Optional) Displays the configuration AAA accounting default methods.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Using AAA Server VSAs

VSAs

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

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

```
protocol : attribute seperator value *
```

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

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

VSA Format

The following VSA protocol options are supported by the Cisco Nexus device:

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

The following attributes are supported by the Cisco Nexus device:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space.
- accountinginfo—Stores additional accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

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

```
shell:roles="roleA roleB ..."
```

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

Note For information on Cisco Unified Wireless Network TACACS+ configurations and to change the user roles, see Cisco Unified Wireless Network TACACS+ Configuration.

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

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

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

For additional information, see the Configuring User Accounts and RBAC chapter in the System Management Configuration Guide for your Cisco Nexus device.

Monitoring and Clearing the Local AAA Accounting Log

The Cisco Nexus device maintains a local log for the AAA accounting activity.

Procedure

	Command or Action	Purpose
Step 1	switch# **show accounting log** [*size*] [**start-time** *year month day hh* **:** *mm* **:** *ss*]	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the size argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a start time for the log output.
Step 2	switch# **clear accounting log**	(Optional) Clears the accounting log contents.

Verifying the AAA Configuration

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

Procedure

	Command or Action	Purpose
Step 1	**show aaa accounting**	Displays AAA accounting configuration.
Step 2	**show aaa authentication** [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
Step 3	**show aaa authorization**	Displays AAA authorization information.
Step 4	**show aaa groups**	Displays the AAA server group configuration.
Step 5	**show running-config aaa** [**all**]	Displays the AAA configuration in the running configuration.
Step 6	**show startup-config aaa**	Displays the AAA configuration in the startup configuration.

Configuration Examples for AAA

The following example shows how to configure AAA:

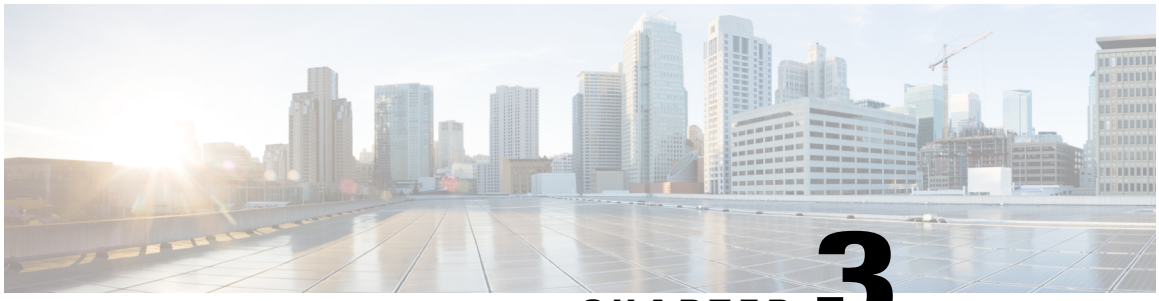
```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

Default AAA Settings

The following table lists the default settings for AAA parameters.

Table 4: Default AAA Parameters

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



CHAPTER 3

Configuring RADIUS

This chapter contains the following sections:

Configuring RADIUS

Information About RADIUS

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

RADIUS Network Environments

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

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

- Networks with multiple-vendor network devices, each supporting RADIUS.

  For example, network devices from several vendors can use a single RADIUS server-based security database.

- Networks already using RADIUS.

  You can add a Cisco Nexus device with RADIUS to the network. This action might be the first step when you make a transition to an AAA server.

- Networks that require resource accounting.

  You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider

(ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.

- Networks that support authentication profiles.

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

Information About RADIUS Operations

When a user attempts to log in and authenticate to a Cisco Nexus device using RADIUS, the following process occurs:

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

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

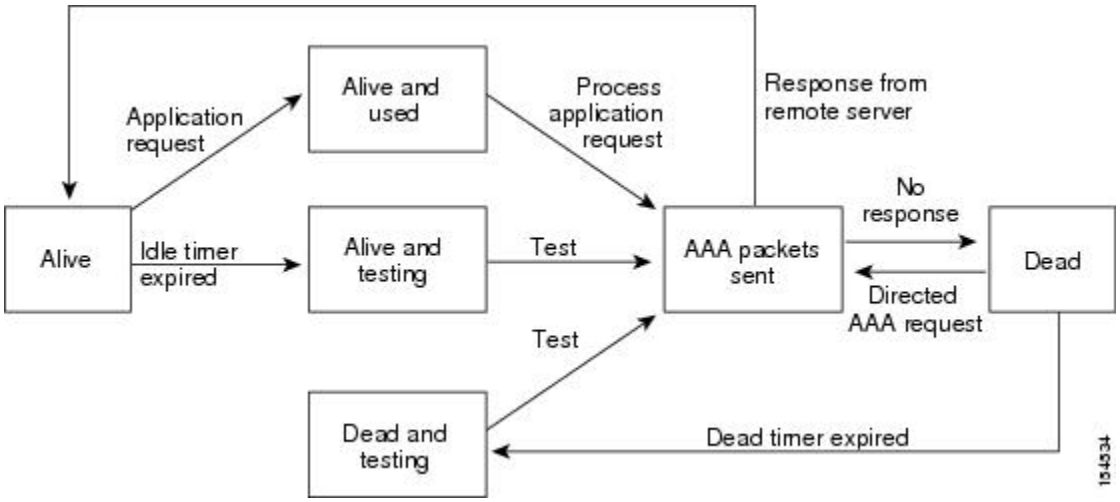
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause delay in processing of AAA requests. You can configure the switch to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The switch marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The switch periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This process verifies that a RADIUS server is in a working state before real AAA requests are sent to the server. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the switch displays an error message that a failure is taking place.

The following figure shows the different RADIUS server states:

Figure 2: 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 vendor-specific attributes (VSAs) between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

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

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

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

The following VSA protocol options are supported by the Cisco Nexus device:

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

The Cisco Nexus device supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white spaces.
- 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 must obtain IPv4 addresses or hostnames for the RADIUS servers.
- You must obtain preshared keys from the RADIUS servers.
- Ensure that the Cisco Nexus device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following configuration guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco Nexus device.

Configuring RADIUS Servers

This section describes how to configure RADIUS servers.

Procedure

Step 1 Establish the RADIUS server connections to the Cisco Nexus device.

Step 2 Configure the preshared secret keys for the RADIUS servers.

Step 3 If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.

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

- Dead-time interval.
- Allow specification of a RADIUS server at login.
- Transmission retry count and timeout interval.
- Accounting and authentication attributes.

Step 5 If needed, configure periodic RADIUS server monitoring.

Configuring RADIUS Server Hosts

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*}	Specifies the IPv4 address or hostname for a RADIUS server.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to configure host 10.10.1.1 as a RADIUS server:

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

Configuring RADIUS Global Preshared Keys

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

Before You Begin

Obtain the preshared key values for the remote RADIUS servers

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server key** [**0** | **7**] *key-value*	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters.

	Command or Action	Purpose
		By default, no preshared key is configured.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure preshared keys at the global level for all servers used by the device:

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

Configuring RADIUS Server Preshared Keys

A preshared key is a shared secret text string between the Cisco Nexus device and the RADIUS server host.

Before You Begin

Obtain the preshared key values for the remote RADIUS servers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	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	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure RADIUS preshared keys:

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

Configuring RADIUS Server Groups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive, alphanumeric string with a maximum of 127 characters.
Step 3	switch (config-radius)# **server** {*ipv4-address* |*server-name*}	Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	switch (config-radius)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.

	Command or Action	Purpose
Step 5	switch(config-radius)# **source-interface** *interface*	(Optional) Assigns a source interface for a specific RADIUS server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip radius source-interface** command.
Step 6	switch(config-radius)# **exit**	Exits configuration mode.
Step 7	switch(config)# **show radius-server group** [*group-name*]	(Optional) Displays the RADIUS server group configuration.
Step 8	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to configure a RADIUS server group:

```
switch# configure terminal
switch (config)# aaa group server radius RadServer
switch (config-radius)# server 10.10.1.1
switch (config-radius)# deadtime 30
switch (config-radius)# use-vrf management
switch (config-radius)# exit
switch (config)# show radius-server group
switch (config)# copy running-config startup-config
```

What to Do Next

Apply the RADIUS server groups to an AAA service.

Configuring the Global Source Interface for RADIUS Server Groups

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. You can also configure a different source interface for a specific RADIUS server group.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip radius source-interface** *interface*	Configures the global source interface for all RADIUS server groups configured on the device. The source interface can be the management or the VLAN interface.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration information.
Step 5	switch# **copy running-config startup config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure the mgmt 0 interface as the global source interface for RADIUS server groups:

```
switch# configure terminal
switch(config)# ip radius source-interface mgmt 0
switch(config)# exit
switch# copy running-config startup-config
```

Allowing Users to Specify a RADIUS Server at Login

You can allow users to specify a RADIUS server at login.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **radius-server directed-request**	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show radius-server directed-request**	(Optional) Displays the directed request configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to allow users to select a RADIUS server when logging in to a network:

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

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the Cisco Nexus device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **radius-server retransmit** *count*	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	switch(config)# **radius-server timeout** *seconds*	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to set the retry count to 3 and the transmission timeout interval to 5 seconds for RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server retransmit 3
switch(config)# radius-server timeout 5
switch(config)# exit
switch# copy running-config startup-config
```

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

By default, a Cisco Nexus switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the switch waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.
Step 3	switch(config)#**radius-server host** {*ipv4-address* | *host-name*} **timeout** *seconds*	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set the RADIUS transmission retry count to 3 and the timeout interval to 10 seconds on RADIUS host server server1:

```
switch# configure terminal
switch(config)# radius-server host server1 retransmit 3
switch(config)# radius-server host server1 timeout 10
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | | *host-name*} **acct-port** *udp-port*	(Optional) Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.

	Command or Action	Purpose
Step 3	switch(config)# **radius-server host** {*ipv4-address* | | *host-name*} **accounting**	(Optional) Specifies that the specified RADIUS server is to be used only for accounting purposes. The default is both accounting and authentication.
Step 4	switch(config)# **radius-server host** {*ipv4-address* | | *host-name*} **auth-port** *udp-port*	(Optional) Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	switch(config)# **radius-server host** {*ipv4-address* | | *host-name*} **authentication**	(Optional) Specifies that the specified RADIUS server only be used for authentication purposes. The default is both accounting and authentication.
Step 6	switch(config)# **exit**	Exits configuration mode.
Step 7	switch(config)# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 8	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure accounting and authentication attributes for a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 acct-port 2004
switch(config)# radius-server host 10.10.1.1 accounting
switch(config)# radius-server host 10.10.2.2 auth-port 2005
switch(config)# radius-server host 10.10.2.2 authentication
switch # exit
switch # copy running-config startup-config
switch #
```

Configuring Periodic RADIUS Server Monitoring

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

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

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

The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the switch does not perform periodic RADIUS server monitoring.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	switch(config)# **radius-server deadtime** *minutes*	Specifies the number of minutes before the switch checks a RADIUS server that was previously unresponsive. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure RADIUS server host 10.10.1.1 with a username (user1) and password (Ur2Gd2BH) and with an idle timer of 3 minutes and a deadtime of 5 minutes:

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

Configuring the Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco Nexus device waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

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

Procedure

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

This example shows how to configure a deadtime of 5 minutes for a radius server:

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

Manually Monitoring RADIUS Servers or Groups

Procedure

	Command or Action	Purpose
Step 1	switch# **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password* **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a RADIUS server to confirm availability.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a RADIUS server group to confirm availability.

This example shows how to send a test message to the RADIUS server and server group to confirm availability:

```
switch# test aaa server radius 10.10.1.1 user 1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Verifying the RADIUS Configuration

Procedure

	Command or Action	Purpose
Step 1	switch# **show running-config radius** [**all**]	Displays the RADIUS configuration in the running configuration.
Step 2	switch# **show startup-config radius**	Displays the RADIUS configuration in the startup configuration.
Step 3	switch# **show radius-server** [*server-name* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

Displaying RADIUS Server Statistics

Procedure

	Command or Action	Purpose
Step 1	switch# **show radius-server statistics** {*hostname* | *ipv4-address*}	Displays the RADIUS statistics.

Clearing RADIUS Server Statistics

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

Before You Begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

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

Configuration Examples for RADIUS

The following example shows how to configure RADIUS:

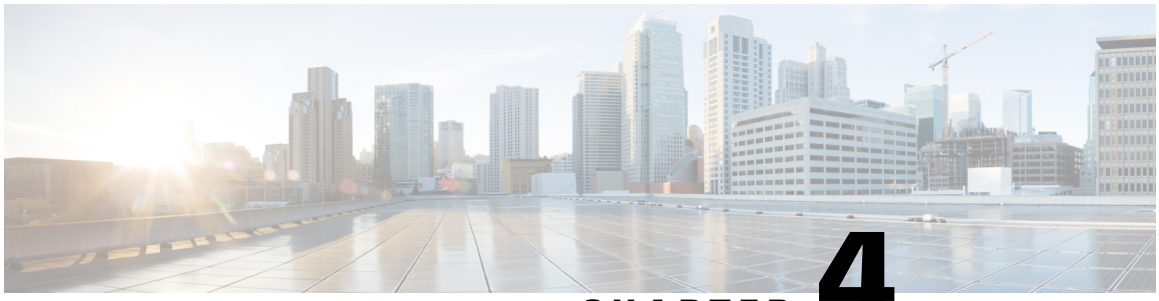
```
switch# configure terminal
switch(config)# radius-server key 7 "ToIkLhPpG"
switch(config)# radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# exit
switch(config-radius)#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

Default Settings for RADIUS

The following table lists the default settings for RADIUS parameters.

Table 5: 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



CHAPTER 4

Configuring TACACS+

This chapter contains the following sections:

About Configuring TACACS+

Information About TACACS+

The Terminal Access Controller Access Control System Plus (TACACS+) security protocol provides centralized validation of users attempting to gain access to a Cisco Nexus device. TACACS+ services are maintained in a database on a TACACS+ daemon typically running 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 Cisco Nexus device are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service (authentication, authorization, and accounting) independently. Each service is associated with 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. The Cisco Nexus device provides centralized authentication using the TACACS+ protocol.

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Nexus device can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

User Login with TACACS+

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco Nexus device using TACACS+, the following actions occur:

1 When the Cisco Nexus device establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.

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

2 The Cisco Nexus device receives one of the following responses from the TACACS+ daemon:

- ACCEPT—User authentication succeeds and service begins. If the Cisco Nexus device requires user authorization, authorization begins.
- REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
- ERROR—An error occurred at some time during authentication dither at the daemon or in the network connection between the daemon and the Cisco Nexus device. If the Cisco Nexus deviceh receives an ERROR response, the switch tries to use an alternative method for authenticating the user.

The user also undergoes an additional authorization phase, if authorization has been enabled on the Cisco Nexus device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3 If TACACS+ authorization is required, the Cisco Nexus device again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

Services include the following:

- ◦ Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
  ◦ Connection parameters, including the host or client IP address (IPv4), access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

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

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

Command Authorization Support for TACACS+ Servers

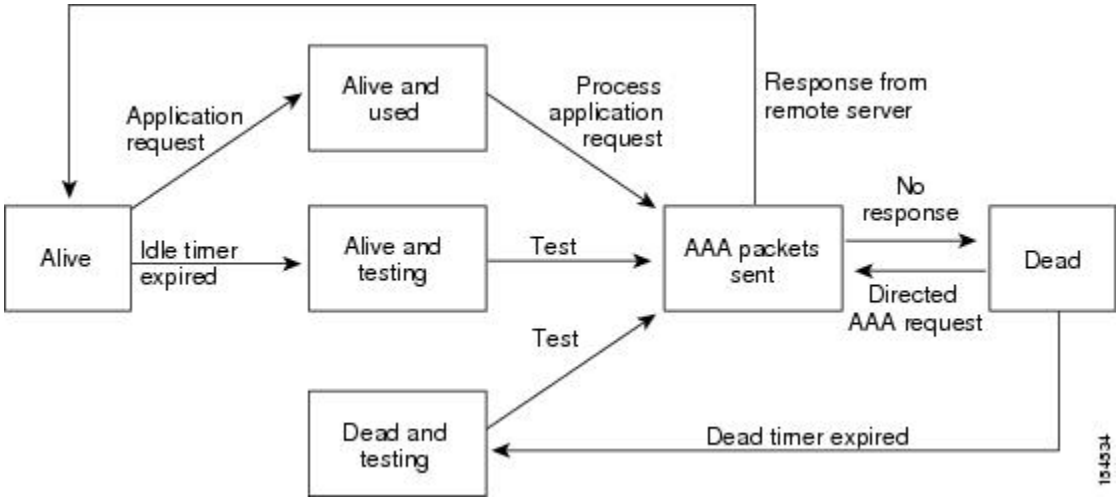
By default, command authorization is done against a local database in the Cisco NX-OS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco Nexus device can periodically monitor an TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco Nexus device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. The Cisco Nexus device periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent to the server. Whenever an TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco Nexus device displays an error message that a failure is taking place before it can impact performance.

The following figure shows the different TACACS+ server states:

Figure 3: TACACS+ Server States



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

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- You must obtain the IPv4 addresses or hostnames for the TACACS+ servers.

- You must obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations for TACACS+

TACACS+ has the following configuration guidelines and limitations:

- You can configure a maximum of 64 TACACS+ servers on the Cisco Nexus device.

Configuring TACACS+

TACACS+ Server Configuration Process

This section describes how to configure TACACS+ servers.

Procedure

Step 1 Enable TACACS+.

Step 2 Establish the TACACS+ server connections to the Cisco Nexus device.

Step 3 Configure the preshared secret keys for the TACACS+ servers.

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

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

- Dead-time interval
- Allow TACACS+ server specification at login
- Timeout interval
- TCP port

Step 6 If needed, configure periodic TACACS+ server monitoring.

Enabling TACACS+

Although by default, the TACACS+ feature is disabled on the Cisco Nexus device. You can enable the TACACS+ feature to access the configuration and verification commands for authentication, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **feature tacacs+**	Enables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

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

If a preshared key is not configured for a configured TACACS+ server, a warning message is issued if a global key is not configured. If a TACACS+ server key is not configured, the global key (if configured) is used for that server.

Before you configure TACACS+ server hosts, you should do the following:

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}	Specifies the IPv4 address or hostname for a TACACS+ server.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

You can delete a TACACS+ server host from a server group.

Configuring TACACS+ Global Preshared Keys

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

Before you configure preshared keys, you should do the following:

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

Procedure

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

The following example shows how to configure global preshared keys:

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

Related Topics

Enabling TACACS+ , on page 40

Configuring TACACS+ Server Preshared Keys

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure the TACACS+ preshared keys:

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

Configuring TACACS+ Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the TACACS+ protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before You Begin

You must use the **feature tacacs+** command to enable TACACS+ before you configure TACACS+.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **aaa group server tacacs+** *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.
Step 3	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 4	switch(config-tacacs+)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	switch(config-tacacs+)# **source-interface** *interface* **Example:** `switch(config-tacacs+)# source-interface mgmt 0`	(Optional) Assigns a source interface for a specific TACACS+ server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip tacacs source-interface** command.
Step 6	switch(config-tacacs+)# **exit**	Exits configuration mode.
Step 7	switch(config)# **show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration.
Step 8	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a TACACS+ server group:

```
switch# configure terminal
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# deadtime 30
switch(config-tacacs+)# exit
switch(config)# show tacacs-server groups
switch(config)# copy running-config startup-config
```

Configuring the Global Source Interface for TACACS+ Server Groups

You can configure a global source interface for TACACS+ server groups to use when accessing TACACS+ servers. You can also configure a different source interface for a specific TACACS+ server group.

Procedure

	Command or Action	Purpose

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

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

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

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

Specifying a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request by enabling the directed-request option. By default, a Cisco Nexus device forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as *username@hostname*, where *hostname* is the name of a configured RADIUS server.

Note User specified logins are only supported for Telnet sessions.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show tacacs-server directed-request**	(Optional) Displays the TACACS+ directed request configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Configuring AAA Authorization on TACACS+ Servers

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

Before You Begin

Enable TACACS+.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}
	
	Example:
	`switch(config)# aaa authorization ssh-certificate`
	`default group TACACSServer1 TACACSServer2`
		Configures the default AAA authorization method for the TACACS+ servers.
		
		The **ssh-certificate** keyword configures TACACS+ or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.
		
		The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.

	Command or Action	Purpose

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

Step 4	**show aaa authorization** [**all**] **Example:** `switch# show aaa authorization`	(Optional) Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Command Authorization on TACACS+ Servers

You can configure authorization for commands on TACACS+ servers. Command authorization disables user role-based authorization control (RBAC), including the default roles.

Note By default, context-sensitive help and command tab completion show only the commands that are supported for a user as defined by the assigned roles. When you enable command authorization, the Cisco NX-OS software displays all commands in the context sensitive help and in tab completion, regardless of the role assigned to the user.

Before You Begin

Enable TACACS+.

Configure TACACS host and server groups before configuring AAA command authorization.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authorization** {**commands** | **config-commands**} **default** [**group** *group-list* [**local**] | **local**]	Configures the default authorization method for commands for all roles. The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword

	Command or Action	Purpose
	Example: `switch(config)# aaa authorization commands default group TacGroup`	configures authorization sources for all configuration commands. The default authorization for all commands is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers that belong to this group are contacted for command authorization. The **local** method uses the local role-based database for authorization. The **local** method is used only if all the configured server groups fail to respond and you have configured **local** as the fallback method. The default method is **local**. If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	**show aaa authorization** [**all**] **Example:** `switch(config)# show aaa authorization`	(Optional) Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Testing Command Authorization on TACACS+ Servers

You can test the command authorization for a user on the TACACS+ servers.

Note You must send correct commands for authorization or the results might not be reliable.

Before You Begin

Enable TACACS+.

Ensure that you have configured command authorization for the TACACS+ servers.

Procedure

	Command or Action	Purpose
Step 1	**test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string* **Example:** `switch# test aaa authorization command-type commands user TestUser command reload`	Tests a user's authorization for a command on the TACACS+ servers. The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands. **Note** Put double quotes (") before and after the *command-string* argument if it contains spaces.

Enabling and Disabling Command Authorization Verification

You can enable and disable command authorization verificaiton on the command-line interface (CLI) for the default user session or for another username.

Note The commands do not execute when you enable authorization verification.

Procedure

	Command or Action	Purpose
Step 1	**terminal verify-only** [**username** *username*] **Example:** `switch# terminal verify-only`	Enables command authorization verification. After you enter this command, the Cisco NX-OS software indicates whether the commands you enter are authorized or not.
Step 2	**terminal no verify-only** [**username** *username*] **Example:** `switch# terminal no verify-only`	Disables command authorization verification.

Configuring Privilege Level Support for Authorization on TACACS+ Servers

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

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

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

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

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

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature privilege** **Example:** `switch(config)# feature privilege`	Enables or disables the cumulative privilege of roles. Users can see the **enable** command only if this feature is enabled. The default is disabled.
Step 3	[**no**] **enable secret** [**0** | **5**] *password* [**priv-lvl** *priv-lvl* | **all**] **Example:** `switch(config)# enable secret 5 def456 priv-lvl 15`	Enables or disables a secret password for a specific privilege level. Users are prompted to enter the correct password upon each privilege level escalation. The default is disabled. You can enter **0** to specify that the password is in clear text or **5** to specify that the password is in encrypted format. The *password* argument can be up to 64 alphanumeric characters. The *priv-lvl* argument is from 1 to 15. **Note** To enable the secret password, you must have enabled the cumulative privilege of roles by entering the **feature privilege** command.

	Command or Action	Purpose
Step 4	[**no**] **username** *username* **priv-lvl** *n* **Example:** `switch(config)# username user2 priv-lvl 15`	Enables or disables a user to use privilege levels for authorization. The default is disabled. The **priv-lvl** keyword specifies the privilege level to which the user is assigned. There is no default privilege level. Privilege levels 0 to 15 (priv-lvl 0 to priv-lvl 15) map to user roles priv-0 to priv-15.
Step 5	**show privilege** **Example:** `switch(config)# show privilege`	(Optional) Displays the username, current privilege level, and status of cumulative privilege support.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

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

Permitting or Denying Commands for Users of Privilege Roles

As a network administrator, you can modify the privilege roles to permit users to execute specific commands or to prevent users from running those commands.

You must follow these guidelines when changing the rules of privilege roles:

- You cannot modify the priv-14 and priv-15 roles.
- You can add deny rules only to the priv-0 role.
- These commands are always permitted for the priv-0 role: **configure**, **copy**, **dir**, **enable**, **ping**, **show**, **ssh**, **telnet**, **terminal**, **traceroute**, **end**, and **exit**.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **role name priv-***n*
	
	Example:
	`switch(config)# role name priv-5`
	`switch(config-role)#`
		Enables or disables a privilege role and enters role configuration mode. The *n* argument specifies the privilege level and is a number between 0 and 13.

Step 3	**rule** *number* {**deny** | **permit**} **command** *command-string* **Example:** `switch(config-role)# rule 2 permit command pwd`	Configures a command rule for users of privilege roles. These rules permit or deny users to execute specific commands. You can configure up to 256 rules for each role. The rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1. The *command-string* argument can contain spaces. **Note** Repeat this command for 256 rules.

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

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

Configuring the Global TACACS+ Timeout Interval

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **tacacs-server timeout** *seconds*	Specifies the timeout interval for TACACS+ servers. The default timeout interval is 5 second and the range is from 1 to 60 seconds.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Configuring the Timeout Interval for a Server

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

Procedure

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

Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, the Cisco Nexus device uses port 49 for all TACACS+ requests.

Procedure

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

The following example shows how to configure TCP ports:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 port 2
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco Nexus device sends out a test packet.You can configure this option to test servers periodically, or you can run a one-time only test.

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

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

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

To configure periodic TACACS+ server monitoring, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes and the valid range is from 0 to 1440 minutes. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the number minutes before the Cisco Nexus device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes and the valid range is 0 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	switch# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure periodic TACACS+ server monitoring:

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

Configuring the Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco Nexus device waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

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

Procedure

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

Manually Monitoring TACACS+ Servers or Groups

Procedure

	Command or Action	Purpose
Step 1	switch# **test aaa server tacacs+** {*ipv4-address* | *host-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a TACACS+ server to confirm availability.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a TACACS+ server group to confirm availability.

The following example shows how to manually issue a test message:

```
switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group TacGroup user2 As3He3CI
```

Disabling TACACS+

You can disable TACACS+.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **no feature tacacs+**	Disables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Displaying TACACS+ Statistics

To display the statistics, the switch maintains for TACACS+ activity, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **show tacacs-server statistics** {*hostname* | *ipv4-address*}	Displays the TACACS+ statistics.

For detailed information about the fields in the output from this command, see the *Command Reference* for your Nexus switch.

Verifying the TACACS+ Configuration

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

Procedure

	Command or Action	Purpose
Step 1	switch# **show tacacs+** {**status** | **pending** | **pending-diff**}	Displays the TACACS+ Cisco Fabric Services distribution status and other details.
Step 2	switch# **show running-config tacacs** [**all**]	Displays the TACACS+ configuration in the running configuration.
Step 3	switch# **show startup-config tacacs**	Displays the TACACS+ configuration in the startup configuration.

	Command or Action	Purpose
Step 4	switch# **show tacacs-serve** [*host-name* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

Configuration Examples for TACACS+

The following example shows how to configure TACACS+:

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

The following example shows how to enable tacacs+ and how to configure the tacacs+ server preshared keys to specify remote AAA servers to authenticate server group TacServer1:

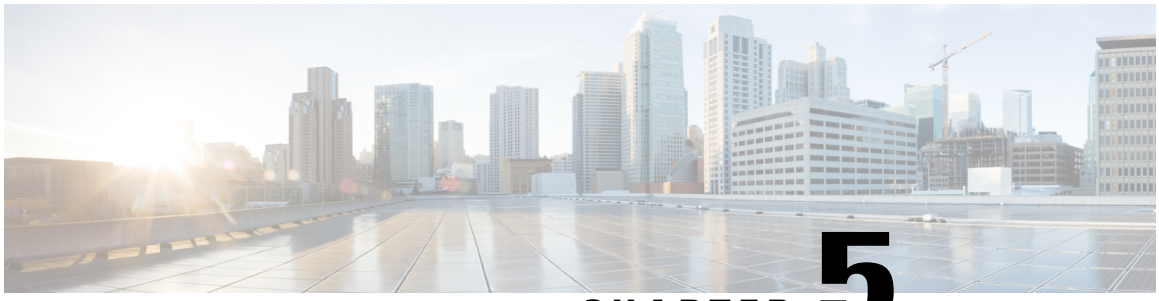
```
switch# config t
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ikvhw10"
switch(config)# tacacs-server host 1.1.1.1
switch(config)# tacacs-server host 1.1.1.2

switch(config)# aaa group server tacacs+ TacServer1
switch(config-tacacs+)#   server 1.1.1.1
switch(config-tacacs+)#   server 1.1.1.2
```

Default Settings for TACACS+

The following table lists the default settings for TACACS+ parameters.

Table 6: Default TACACS+ Parameters

Parameters	Default
TACACS+	Disabled
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test



CHAPTER 5

Configuring SSH and Telnet

This chapter contains the following sections:

Configuring SSH and Telnet

Information About SSH and Telnet

SSH Server

The Secure Shell Protocol (SSH) server feature enables a SSH client to make a secure, encrypted connection to a Cisco Nexus device. SSH uses strong encryption for authentication. The SSH server in the Cisco Nexus device switch interoperates with publicly and commercially available SSH clients.

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

SSH Client

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

The SSH client in the Cisco Nexus device works with publicly and commercially available SSH servers.

SSH Server Keys

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

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography

- SSH version 2 using the Digital System Algrorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts three types of key-pairs for use by SSH version 2:

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

By default, the Cisco Nexus device generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)

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

Telnet Server

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

The Telnet server is enabled by default on the Cisco Nexus device.

Guidelines and Limitations for SSH

SSH has the following configuration guidelines and limitations:

- The Cisco Nexus device supports only SSH version 2 (SSHv2).

Configuring SSH

Generating SSH Server Keys

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

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key. The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	switch# **show ssh key**	(Optional) Displays the SSH server keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to generate an SSH server key:

```
switch# configure terminal
switch(config)# ssh key rsa 2048
switch(config)# exit
switch# show ssh key
switch# copy running-config startup-config
```

Specifying the SSH Public Keys for User Accounts

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

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Specifying the SSH Public Keys in Open SSH Format

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in SSH format.
Step 3	switch(config)# **exit**	Exits global configuration mode.

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

The following example shows how to specify an SSH public key in open SSH format:

```
switch# configure terminal
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Note The **username** command in the example above is a single line that has been broken for legibility.

Specifying the SSH Public Keys in IETF SECSH Format

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

Procedure

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

The following example shows how to specify the SSH public key in the IETF SECSH format:

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Specifying the SSH Public Keys in PEM-Formatted Public Key Certificate Form

You can specify the SSH public keys in PEM-formatted Public Key Certificate form for user accounts.

Procedure

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP
Step 2	switch# **configure terminal**	Enters configuration mode.
Step 3	switch# **show user-account**	(Optional) Displays the user account configuration.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to specify the SSH public keys in PEM-formatted public key certificate form:

```
switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem
switch# configure terminal
switch# show user-account
switch# copy running-config startup-config
```

Starting SSH Sessions to Remote Devices

You can start SSH sessions to connect to remote devices from your Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]	Creates an SSH session to a remote device. The *hostname* argument can be an IPv4 address or a hostname.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, you establish a trusted SSH relationship with that server.

Procedure

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

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server. The default is enabled.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	switch# **show ssh server**	(Optional) Displays the SSH server configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

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

Note To reenable SSH, you must first generate an SSH server key.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **no feature ssh**	Disables the SSH server.
Step 3	switch(config)# **no ssh key** [**dsa** | **rsa**]	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	switch# **show ssh key**	(Optional) Displays the SSH server configuration.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

Procedure

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

Configuration Examples for SSH

The following example shows how to configure SSH:

Procedure

Step 1 Generate an SSH server key.

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

Step 2 Enable the SSH server.

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

Note This step should not be required because the SSH server is enabled by default.

Step 3 Display the SSH server key.

```
switch(config)# show ssh key
rsa Keys generated:Fri May  8 22:09:47 2009

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYzCfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZ/
cTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4ZXIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5/
Ninn0Mc=

bitcount:1024

fingerprint:
4b:4d:f6:b9:42:e9:d9:71:3c:bd:09:94:4a:93:ac:ca
*************************************
could not retrieve dsa key information
*************************************
```

Step 4 Specify the SSH public key in Open SSH format.

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
```

Step 5 Save the configuration.

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

Configuring Telnet

Enabling the Telnet Server

By default, the Telnet server is enabled. You can disable the Telnet server on your Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **feature telnet**	Disables the Telnet server. The default is enabled.

Reenabling the Telnet Server

If the Telnet server on your Cisco Nexus device has been disabled, you can reenable it.

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **feature telnet**	Reenables the Telnet server.

Starting Telnet Sessions to Remote Devices

Before you start a Telnet session to connect to remote devices, you should do the following:

- Obtain the hostname for the remote device and, if needed, obtain the username on the remote device.
- Enable the Telnet server on the Cisco Nexus device.
- Enable the Telnet server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet** *hostname*	Creates a Telnet session to a remote device. The *hostname* argument can be an IPv4 address or a device name.

The following example shows how to start a Telnet session to connect to a remote device:

```
switch# telnet 10.10.1.1
Trying 10.10.1.1...
Connected to 10.10.1.1.
Escape character is '^]'.
switch login:
```

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco Nexus device.

Procedure

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

Verifying the SSH and Telnet Configuration

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

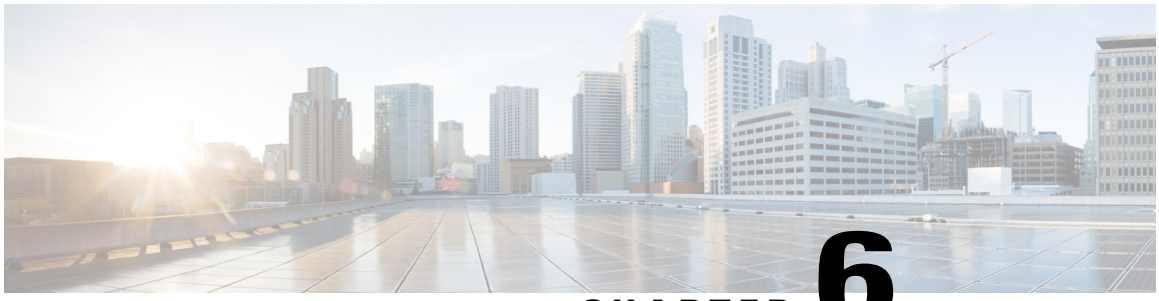
- switch# **show ssh key** [**dsa** | **rsa**]
  Displays SSH server key-pair information.
- switch# **show running-config security** [**all**]
  Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts.
- switch# **show ssh server**
  Displays the SSH server configuration.
- switch# **show user-account**
  Displays user account information.

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 7: Default SSH Parameters

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



CHAPTER 6

Configuring Access Control Lists

This chapter contains the following sections:

Information About ACLs

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

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

IP ACL Types and Applications

The Cisco Nexus device supports IPv4 for security traffic filtering. The switch allows you to use IP access control lists (ACLs) as port ACLs, VLAN ACLs, and Router ACLs as shown in the following table.

Table 8: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	IPv4 ACLs
Router ACL	• VLAN interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	IPv4 ACLs
VLAN ACL (VACL)	An ACL is a VACL when you use an access map to associate the ACL with an action and then apply the map to a VLAN.	IPv4 ACLs
VTY ACL	VTYs	IPv4 ACLs

Application Order

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

1. Port ACL
2. Ingress VACL
3. Ingress Router ACL
4. Egress Router ACL
5. Egress VACL

Rules

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The switch 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.

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.

Protocols

IPv4 and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 ACL, you can specify ICMP by name.

You can specify any protocol by the integer that represents the Internet protocol number.

Implicit Rules

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the switch applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the switch denies unmatched IP traffic.

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

All MAC ACLs include the following implicit rule:

`deny any any` *protocol*

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

Additional Filtering Options

You can identify traffic by using additional options. IPv4 ACLs support the following additional filtering options:

- Layer 4 protocol
- TCP and UDP ports
- ICMP types and codes
- IGMP types
- Precedence level
- Differentiated Services Code Point (DSCP) value

- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections

Sequence Numbers

The Cisco Nexus device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.
- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

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

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

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

In addition, the device allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers.

The Cisco Nexus device stores operator-operand couples in registers called logical operation units (LOUs) to perform operations (greater than, less than, not equal to, and range) on the TCP and UDP ports specified in an IP ACL.

Note The range operator is inclusive of boundary values.

These LOUs minimize the number of ternary content addressable memory (TCAM) entries needed to perform these operations. A maximum of two LOUs are allowed for each feature on an interface. For example an ingress RACL can use two LOUs, and a QoS feature can use two LOUs. If an ACL feature requires more than two arithmetic operations, the first two operations use LOUs, and the remaining access control entries get expanded.

The following guidelines determine when the device stores operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples "gt 10" and "gt 11" would be stored separately in half an LOU each. The couples "gt 10" and "lt 10" would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

The IPv4 TCAMs are single wide.

TCAM region sizes have the following guidelines and limitations:

- To revert to the default ACL TCAM size, use the **no hardware profile tcam region** command. You no longer need to use the **write erase command** and reload the switch.
- Depending on the Cisco Nexus device, each TCAM region might have a different minimum/maximum/aggregate size restriction.
- The default size of the ARPACL TCAM is zero. Before you use the ARP ACLs in a Control Policing Plane (CoPP) policy, you must set the size of this TCAM to a non-zero size.
- You must set the VACL and egress VLAN ACL (E-VACL) size to the same value.
- The total TCAM depth is 2000 for ingress and 1000 for egress, which can be carved in 256 entries blocks.
- After TCAM carving, you must reload the switch.
- All existing TCAMs cannot be set to size 0.
- By default, all IPv6 TCAMs are disabled (the TCAM size is set to 0).

Table 9: TCAM Sizes by ACL Region

TCAM ACL Region	Default Size	Minimum Size	Incremental Size	Maximum Size
SUP (ingress)	128 x 2	128 x 2	N/A	128 x 2

TCAM ACL Region	Default Size	Minimum Size	Incremental Size	Maximum Size
PACL (ingress)	384	ARPACL disabled = 128 ARPACL enabled = 256	256	1664 (combined)
VACL (ingress)	512	0	256	
RACL (ingress)	512	256	256	
QOS (ingress)	256	256	256	
E-VACL (egress)	512	0	256	1024 (combined)
E-RACL (egress)	512	0	256	
NAT	256	256	16	

Licensing Requirements for ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	No license is required to use ACLs.

Prerequisites for 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.

VACLs have the following prerequisite:

- Ensure that the IP ACL that you want to use in the VACL exists and is configured to filter traffic in the manner that you need for this application.

Guidelines and Limitations for ACLs

IP ACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available

prior to committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.

- Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
- IPv4 packets that have IP options (additional IP packet header fields following the destination address field).
- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally.
- To use the **match-local-traffic** option for all inbound and outbound traffic you must first enable the ACL in the software.

VACLs have the following configuration

- We recommend that you perform ACL configurations using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.
- ACL statistics are not supported if the DHCP snooping feature is enabled.
- If an IPv4 ACL, applied as a VLAN ACL, contains one or more ACEs with logical operators for TCP/UDP port numbers, the port numbers are matched in the ingress direction but ignored in the egress direction.
- One VLAN access map can match only one IP ACL.
- An IP ACL can have multiple permit/deny ACEs.
- One VLAN can have only one access map applied.
- Egress RACLs and VACLs cannot be applied in warp mode.
- Egress ACLs cannot be applied to multicast traffic.
- Ingress RACLs cannot distinguish multicast routed versus bridged traffic.
- Link Local (ARP, HSRP, VRRP, OSPF, IGMP, and so on) and IP-option packets cannot be matched in an interface policy, RACL, VACL, or PACL.

Default ACL Settings

The following table lists the default settings for IP ACLs parameters.

Table 10: Default IP ACLs Parameters

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

The following table lists the default settings for VACL parameters.

Table 11: Default VACL Parameters

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

Configuring IP ACLs

Creating an IP ACL

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

Procedure

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

The following example shows how to create an IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list acl-01
```

```
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics
```

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	switch(config)#**ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 4	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 5	switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 6	switch(config-acl)# [**no**] **statistics**	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the ACL. The **no** option stops the switch from maintaining global statistics for the ACL.
Step 7	switch#**show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 8	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 78

Removing an IP ACL

You can remove an IP ACL from the switch.

Before you remove an IP ACL from the switch, be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 4	switch# **show running-config**	(Optional) Displays the ACL configuration. The removed IP ACL should not appear.
Step 5	switch# **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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the

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

Applying an IP ACL to mgmt0

You can apply an IPv4 ACL to the management interface (mgmt0).

Before You Begin

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

Procedure

	Command or Action	Purpose

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

Step 2	**interface mgmt** *port*
	
	Example:
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Enters configuration mode for the management interface.

Step 3	**ip access-group** *access-list* {**in** | **out**} **Example:** `switch(config-if)#ip access-group acl-120 out`	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	**show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	(Optional) Displays the ACL configuration.

	Command or Action	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.

Related Topics

- Creating an IP ACL

Applying an IP ACL as a Port ACL

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

Note Some configuration parameters when applied to an PortChannel are not reflected on the configuration of the member ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** [*chassis*/]*slot*/*port* | **port-channel** *channel-number*}	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **ip port access-group** *access-list* **in**	Applies an IPv4 ACL to the interface or PortChannel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 4	switch# **show running-config**	(Optional) Displays the ACL configuration.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Logical operation units (LOUs) are not available for router ACLs applied in the out direction. If an IPv4 ACL is applied as a router ACL in the out direction, access control entiries (ACEs) that contain logical operators for TCP/UDP port numbers are expanded internally to multiple ACEs and might require more TCAM entries when compared to the same ACL applied in the in direction.

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port*[**.** *number*] • switch(config)# **interface port-channel** *channel-number*[**.** *number*] • switch(config)# **interface tunnel** *tunnel-number* • switch(config)# **interface vlan** *vlan-ID* • switch(config)# **interface mgmt** *port*	Enters configuration mode for the interface type that you specified.
Step 3	switch(config-if)# **ip access-group** *access-list* {**in** | **out**}	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	switch(config-if)# **show running-config aclmgr**	(Optional) Displays the ACL configuration.

	Command or Action	Purpose
Step 5	switch(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, perform one of the following tasks:

- switch# **show running-config**
  Displays ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
- switch# **show running-config interface**
  Displays the configuration of an interface to which you have applied an ACL.

For detailed information about the fields in the output from these commands, refer to the *Command Reference* for your Cisco Nexus device.

Monitoring and Clearing IP ACL Statistics

Use the **show ip access-lists** command to display statistics about an IP ACL, including the number of packets that have matched each rule. For detailed information about the fields in the output from this command, see the *Command Reference* for your Cisco Nexus device.

Note The mac access-list is applicable to non-IPv4 traffic only.

- switch# **show ip access-lists** *name*
  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.
- switch#**show ip access-lists** *name*
  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.
- switch# **clear ip access-list counters** [*access-list-name*]
  Clears statistics for all IP ACLs or for a specific IP ACL.
- switch# **clear ip access-list counters** [*access-list-name*]
  Clears statistics for all IP ACLs or for a specific IP ACL.

Information About VLAN ACLs

A VLAN ACL (VACL) is one application of a IP ACL. You can configure VACLs to apply to all packets that are bridged within a VLAN. VACLs are used strictly for security packet filtering. VACLs are not defined by direction (ingress or egress).

VACLs and Access Maps

VACLs use access maps to link an IP ACL to an action. The switch takes the configured action on packets permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.
- Drop—Drops the traffic.

Statistics

The Cisco Nexus device can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

Note The Cisco Nexus device does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the switch maintains statistics for that VACL. This allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

Configuring VACLs

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP ACL with an action to be applied to the matching traffic.

To create or change a VACL, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vlan access-map** *map-name*	Enters access map configuration mode for the access map specified.
Step 3	switch(config-access-map)# **match ip address** *ip-access-list*	Specifies an IPv4 ACL for the map.
Step 4	switch(config-access-map)# **action** {**drop** | **forward**}	Specifies the action that the switch applies to traffic that matches the ACL.
Step 5	switch(config-access-map)# [**no**] **statistics**	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the VACL. The **no** option stops the switch from maintaining global statistics for the VACL.
Step 6	switch(config-access-map)# **show running-config**	(Optional) Displays ACL configuration.
Step 7	switch(config-access-map)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

Be sure that you know whether the VACL is applied to a VLAN. The switch allows you to remove VACLs that are current applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the switch considers the removed VACL to be empty.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **no vlan access-map** *map-name*	Removes the VLAN access map configuration for the specified access map.
Step 3	switch(config)# **show running-config**	(Optional) Displays ACL configuration.

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

Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# [**no**] **vlan filter** *map-name* **vlan-list** *list*	Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. The **vlan-list** command can specify a list of up to 32 VLANs, but multiple **vlan-list** commands can be configured to cover more than 32 VLANs.
Step 3	switch(config)# **show running-config**	(Optional) Displays ACL configuration.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Verifying VACL Configuration

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

- switch# **show running-config aclmgr**
  Displays ACL configuration, including VACL-related configuration.
- switch# **show vlan filter**
  Displays information about VACLs that are applied to a VLAN.
- switch# **show vlan access-map**
  Displays information about VLAN access maps.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, perform one of the following tasks:

- switch# **show vlan access-list**
  Displays VACL configuration. If the VLAN access-map includes the **statistics** command, then the **show vlan access-list** command output includes the number of packets that have matched each rule.
- switch# **clear vlan access-list counters**
  Clears statistics for all VACLs or for a specific VACL.

Configuration Examples for VACL

The following example shows how to configure a VACL to forward traffic permitted by an IP ACL named acl-ip-01 and how to apply the VACL to VLANs 50 through 82:

```
switch# configure terminal
switch(config)# vlan access-map acl-ip-map
switch(config-access-map)# match ip address acl-ip-01
switch(config-access-map)# action forward
switch(config-access-map)# exit
 switch(config)# vlan filter acl-ip-map vlan-list 50-82
```

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Procedure

	Command or Action	Purpose

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

Step 2	**hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **ipsg** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Changes the ACL TCAM region size. • **arpacl—**Configures the size of the Address Resolution Protocol (ARP) ACL (ARPACL) TCAM region. • **e-racl—**Configures the size of the egress router ACL (ERACL) TCAM region. • **e-vacl—**Configures the size of the egress VLAN ACL (EVACL) TCAM region. • **ifacl—**Configures the size of the interface ACL (ifacl) TCAM region. • **ipsg—**Configures the size of the IP Source Guard (IPSG) TCAM region. • **nat—**Configures the size of the NAT TCAM region. • **qos—**Configures the size of the quality of service (QoS) TCAM region.

	Command or Action	Purpose
		• **qoslbl—**Configures the size of the QoS Label (qoslbl) TCAM region. • **racl—**Configures the size of the router ACL (RACL) TCAM region. • **vacl—**Configures the size of the VLAN ACL (VACL) TCAM region. • *tcam_size*—TCAM size. The range is from 0 to 2,14,74, 83, 647 entries. **Note** **vacl** and **e-vacl** TCAM regions should be set to the same size.
Step 3	**copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **show hardware profile tcam region**	Displays the TCAM sizes that will be applicable on the next reload of the switch.
Step 5	switch(config)# **reload**	Copies the running configuration to the startup configuration. **Note** The new size values are effective only upon the next reload after saving the **copy running-config to startup-config**.

The following example shows how to change the size of the RACL TCAM region:

```
switch(config)# hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

The following example shows the error message you see when you set the ARP ACL TCAM value to a value other than 0 or 128, and then shows how to change the size of the ARP ACL TCAM region:

```
switch(config)# hardware profile tcam region arpacl 200
ARPACL size can be either 0 or 128

switch(config)# hardware profile tcam region arpacl 128
To start using ARPACL tcam, IFACL tcam size needs to be changed.
Changing IFACL tcam size to 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
```

The following example shows how to configure the TCAM VLAN ACLs on a switch:

```
switch# configure sync
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-sync)# switch-profile s5010
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)# hardware profile tcam region vacl 512
switch(config-sync-sp)# hardware profile tcam region e-vacl 512
switch(config-sync-sp)#
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware profile tcam region
         sup size =   16
        vacl size =  640
       ifacl size =  496
         qos size =  256
       rbacl size =    0
        span size =    0
        racl size = 1536
      e-racl size =  256
      e-vacl size =  640
      qoslbl size =    0
        ipsg size =    0
      arpacl size =    0
   ipv6-racl size =    0
 ipv6-e-racl size =    0
    ipv6-sup size =    0
    ipv6-qos size =    0
         nat size =  256
```

Reverting to the Default TCAM Region Sizes

Procedure

	Command or Action	Purpose

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

Step 2	switch(config)# **no hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **ipsg** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Reverts the configuration to the default ACL TCAM size.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **reload**	Reloads the switch.

The following example shows how to revert to the default RACL TCAM region sizes:

```
switch(config)# no hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-configur startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configuring ACLs on Virtual Terminal Lines

To restrict incoming and outgoing connections for IPv4 between a Virtual Terminal (VTY) line and the addresses in an access list, use the **access-class** command in line configuration mode. To remove access restrictions, use the **no** form of this command.

Follow these guidelines when configuring ACLs on VTY lines:

- Set identical restrictions on all VTY lines because a user can connect to any of them.
- Statistics per entry is not supported for ACLs on VTY lines.

Before You Begin

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

Procedure

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

Step 2	switch(config)# **line vty**
	
	Example:
	`switch(config)# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

Step 3	switch(config-line)# **access-class access-list-number {in | out}**
	
	Example:
	`switch(config-line)# access-class ozi2 in`
	`switch(config-line)#access-class ozi3 out`
	`switch(config)#`
		Specifies inbound or outbound access restrictions.

Step 4	switch(config-line)# **no access-class access-list-number {in | out}**
	
	Example:
	`switch(config-line)# no access-class ozi2 in`
	`switch(config-line)# no access-class ozi3 out`
	`switch(config)#`
		(Optional)
		Removes inbound or outbound access restrictions.

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

Step 6	switch# **show running-config aclmgr** **Example:** `switch# show running-config aclmgr`	(Optional) Displays the running configuration of the ACLs on the switch.

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

The following example shows how to apply the access-class ozi2 command to the in-direction of the vty line.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# line vty
switch(config-line)# access-class ozi2 in
switch(config-line)# exit
switch#
```

Verifying ACLs on VTY Lines

To display the ACL configurations on VTY lines, perform one of the following tasks:

Command	Purpose
show running-config aclmgr	Displays the running configuration of the ACLs configured on the switch.
show users	Displays the users that are connected.
show access-lists *access-list-name*	Display the statistics per entry.

Configuration Examples for ACLs on VTY Lines

The following example shows the connected users on the console line (ttyS0) and the VTY lines (pts/0 and pts/1).

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    ttyS0        Aug 27 20:45   .         14425 *
admin    pts/0        Aug 27 20:06 00:46       14176 (172.18.217.82) session=ssh
admin    pts/1        Aug 27 20:52   .         14584 (10.55.144.118)
```

The following example shows how to allow vty connections to all IPv4 hosts except 172.18.217.82 and how to deny vty connections to any IPv4 host except 10.55.144.118, 172.18.217.79, 172.18.217.82, 172.18.217.92:

```
switch# show running-config aclmgr
!Time: Fri Aug 27 22:01:09 2010
version 5.0(2)N1(1)
ip access-list ozi
  10 deny ip 172.18.217.82/32 any
  20 permit ip any any
ip access-list ozi2
  10 permit ip 10.55.144.118/32 any
  20 permit ip 172.18.217.79/32 any
  30 permit ip 172.18.217.82/32 any
  40 permit ip 172.18.217.92/32 any


line vty
  access-class ozi in
```

```
  access-class ozi2 out
```

The following example shows how to configure the IP access list by enabling per-entry statistics for the ACL:

```
switch# conf t
Enter configuration commands, one per line.
End with CNTL/Z.
switch(config)# ip access-list ozi2
switch(config-acl)# statistics per-entry
switch(config-acl)# deny tcp 172.18.217.83/32 any
switch(config-acl)# exit

switch(config)# ip access-list ozi
switch(config-acl)# statistics per-entry
switch(config-acl)# permit ip 172.18.217.20/24 any
switch(config-acl)# exit
switch#
```

The following example shows how to apply the ACLs on VTY in and out directions:

```
switch(config)# line vty
switch(config-line)# ip access-class ozi in
switch(config-line)# access-class ozi2 out
switch(config-line)# exit
switch#
```

The following example shows how to remove the access restrictions on the VTY line:

```
switch# conf t
Enter configuration commands, one per line. End
with CNTL/Z.
switch(config)# line vty
switch(config-line)# no access-class ozi2 in
switch(config-line)# no ip access-class ozi2 in
switch(config-line)# exit
switch#
```

INDEX

E

G

I

L

M

P

R

S

T

U

V