Cisco Nexus 1000V InterCloud Security Configuration Guide, Release 5.2(1)IC1(1.1)

First Published: July 02, 2013

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-29150-01

CONTENTS

Preface

This preface contains the following sections:

Audience

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

This guide is for network and server administrators with the following experience and knowledge:

- An understanding of virtualization
- Using VMM software to create a virtual machine and configure a VMware vSwitch
- Ability to create an account on provider cloud such as Amazon Web Services (AWS).
- Knowledge of VMware vNetwork Distributed Switch is not required.

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).

Convention	Description
[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.
[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 Cisco Nexus 1000V InterCloud

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

http://www.cisco.com/en/US/partner/products/ps12904/tsd_products_support_series_home.html

General Information

Cisco Nexus 1000V InterCloud Release Notes

Install and Upgrade

Cisco Nexus 1000V InterCloud Installation Guide

Configuration Guides

Cisco Nexus 1000V InterCloud License Configuration Guide

Cisco Nexus 1000V InterCloud High Availability and Redundancy Configuration Guide

Cisco Nexus 1000V InterCloud Interface Configuration Guide

Cisco Nexus 1000V InterCloud Layer 2 Configuration Guide

Cisco Nexus 1000V InterCloud Port Profile Configuration Guide

Cisco Nexus 1000V InterCloud Security Configuration Guide

Cisco Nexus 1000V InterCloud System Management Configuration Guide

Reference Guides

Cisco Nexus 1000V InterCloud Command Reference

Cisco Nexus 1000V InterCloud Verified Scalability Reference

Cisco Nexus 1000V MIB Quick Reference

Troubleshooting and Alerts

Cisco Nexus 1000V Password Recovery Procedure

Cisco Nexus 1000V Documentation

Cisco Nexus 1000V for VMware vSphere Documentation

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

Cisco Prime Network Services Controller Documentation

http://www.cisco.com/en/US/products/ps13213/tsd_products_support_series_home.html

Documentation Feedback

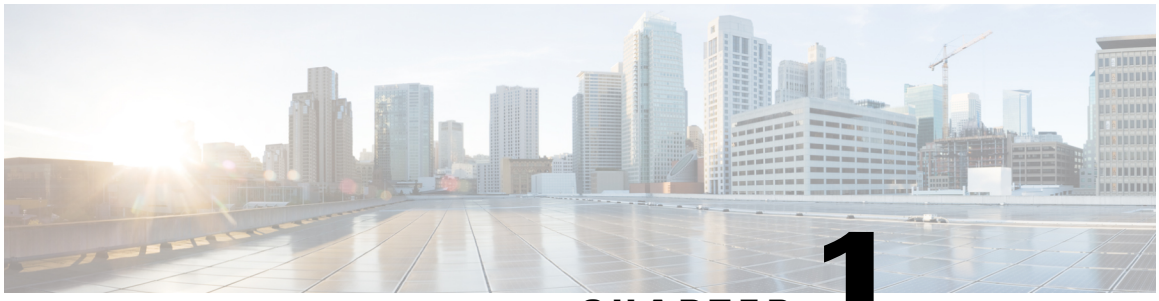
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus1k-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

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

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

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



CHAPTER 1

Overview

This chapter contains the following sections:

User Accounts

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

Authentication, Authorization, and Accounting

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

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate

user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

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

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

RADIUS Security Protocol

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

TACACS+ Security Protocol

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

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

SSH

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

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

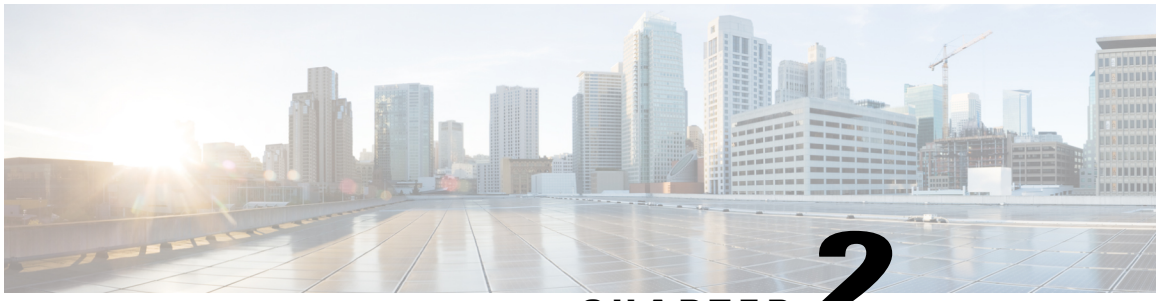
Telnet

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

Access Control Lists

IP ACLs

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



CHAPTER 2

Managing User Accounts

This chapter contains the following sections:

Information About User Accounts

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

- Role
- Username
- Password
- Expiration date

Role

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

```
role: network-admin
  description: Predefined network admin role has access to all commands
  on the switch
```

```
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  1       permit  read-write

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

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

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

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

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

Username

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

Password

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

The following characters are not permitted in clear text passwords:

- dollar signs ($)
- spaces
- equal to sign (=)

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

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

The following table lists the characteristics of strong passwords.

Table 1: Characteristics of Strong Passwords

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

Some examples of strong passwords are as follows:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Check of Password Strength

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

```
switch# config terminal
switch (config)# username daveGreen password davey
password is weak
Password should contain characters from at least three of the classes:
 lower case letters, upper case letters, digits, and special characters
```

Password strength checking can be disabled.

Expiration Date

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

Guidelines and Limitations for Creating User Accounts

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

- You can assign a maximum of 64 user roles to a user account.
- If you have a user account that has the same name as a remote user account on an AAA server, the user roles for the local user account are applied to the remote user, not the user roles configured on the AAA server.

Guidelines for Creating User Accounts

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

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

- You can add a user password as either clear text or encrypted.
  - Clear text passwords are encrypted before they are saved to the running configuration.
  - Encrypted passwords are saved to the running configuration without further encryption.
- A user account can have up to 64 roles, but must have at least one role. For more information about roles, Guidelines for Creating User Accounts, on page 8
- If you do not specify a password, the user might not be able to log in
- For information about using SSH public keys instead of passwords, see Configuring an OpenSSH Key, on page 68.

Default Settings for User Access

Parameters	Default
User account password	Undefined
User account expiration date	None
User account role	Network-operator

Parameters	Default
Interface policy	All interfaces are accessible
VLAN policy	All VLANs are accessible

Configuring User Access

Enabling the Check of Password Strength

Use this procedure to enable the Cisco Nexus 1000V to check the strength of passwords to avoid creating weak passwords for user accounts.

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

Before You Begin

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

Procedure

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

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

Disabling the Check of Password Strength

Before You Begin

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

Procedure

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

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

Creating a User Account

Before You Begin

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

Procedure

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

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

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

Creating a Role

Before You Begin

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- You can configure up to 64 user roles.
- You can configure up to up to 256 rules for each role.
- You can assign a single role to more than one user.
- The rule number specifies the order in which it is applied, in descending order. For example, if a role has three rules, rule 3 is applied first, rule 2 is applied next, and rule 1 is applied last.

- By default, the user roles that you create allow access only to the show, exit, end, and configure terminal commands. You must add rules to allow users to configure features.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role name** *role-name*	Names a user role and places you in role configuration mode for that role. The *role-name* is a case-sensitive, alphanumeric string of up to 16 characters.
Step 3	switch(config-role)# **description** *description-string*	(Optional) Configures the role description, which can include spaces.
Step 4	switch(config-role)# **rule** *number* {**deny**| **permit**} **command** *command-string* • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} Creates one rule to permit or deny all operations. • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name* Creates a rule for feature access. Use the **show role feature** command to display a list of available features. • switch(config-role)# **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name* Creates a rule for feature group access. Use the **show role feature-group** command to display a list of feature groups. **Example:** This example configures a rule that denies access to the **clear users** command.	Creates a rule to permit or deny a specific command. The command you specify can contain spaces and regular expressions.
Step 5	Repeat Step 4 to create all needed rules for the specified role.	
Step 6	switch(config-role)# **show role**	(Optional) Displays the user role configuration.

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

```
switch# configure terminal
switch(config)# role name UserA
switch(config-role)# description Prohibits use of clear commands
switch(config-role)# rule 1 deny command clear users
switch(config-role)# rule 2 deny read-write

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

Creating a Feature Group

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

Before You Begin

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- You can create up to 64 custom feature groups.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **role feature-group name** *group-name*	Places you into the role feature group configuration mode for the named group. group-name—A case-sensitive, alphanumeric string of up to 32 characters in length.
Step 3	switch(config-role-featuregrp)# **show role feature**	Displays a list of available features for use in defining the feature group.
Step 4	switch(config-role-featuregrp)# **feature** *feature-name*	Adds a feature to the feature group. Repeat this step for all features to be added to the feature group.
Step 5	switch(config-role-featuregrp)# **show role feature-group**	(Optional) Displays the feature group configuration.

	Command or Action	Purpose
Step 6	switch(config-role-featuregrp)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

Configuring Interface Access

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

Before You Begin

Before beginning this procedure you must have done the following:

- Logged in to the CLI in EXEC mode
- Created one or more user roles. In this procedure, you will be modifying a role you have already created.

Procedure

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

Configuring VLAN Access

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

Before You Begin

Before beginning this procedure, you must:

- Be logged in to the CLI in EXEC mode.
- Have already created one or more user roles. In this procedure, you will be modifying a role you have already created.

Procedure

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

Verifying the User Access Configuration

Use one of the following commands to verify the configuration.

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

Configuration Examples

Configuration Example for Creating a Feature Group

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

Configuration Example for Creating a Role

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

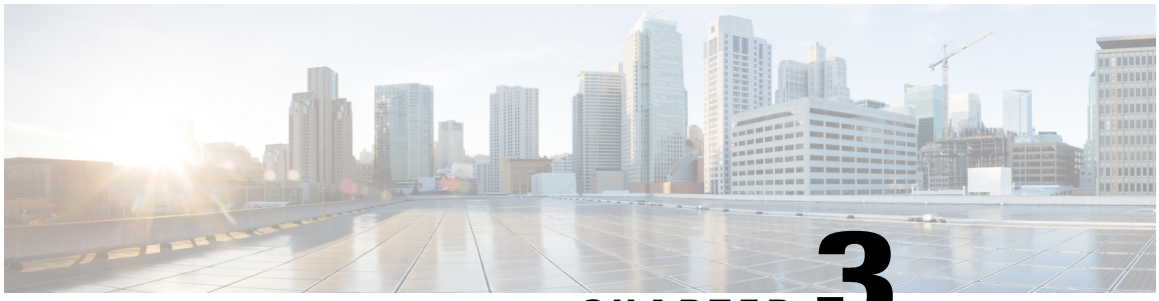
MIBs

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

Feature History for User Accounts

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

Feature Name	Releases	Feature Information
User Accounts	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 3

Configuring AAA

This chapter contains the following sections:

Information about AAA

AAA Security Services

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

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

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

Separate AAA configurations are made for the following services:

- User Telnet or Secure Shell (SSH) login authentication

- Console login authentication
- User management session accounting

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

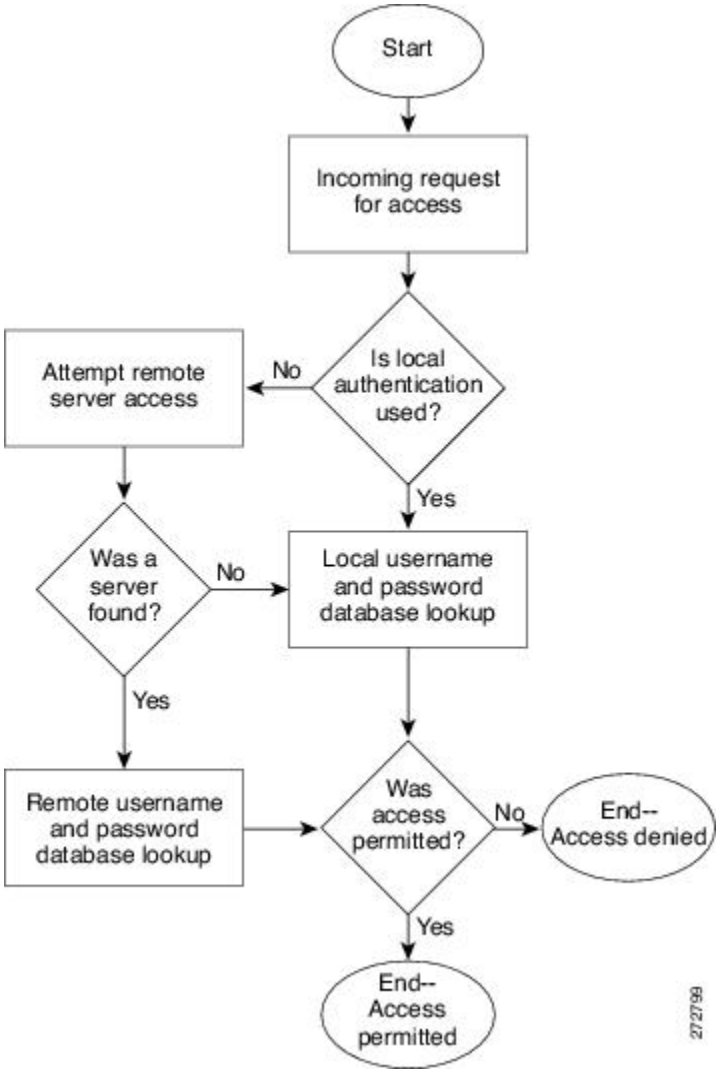
Authentication

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

Authentication is accomplished as follows:

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

Figure 1: Authenticating User Login



Authorization

Authorization restricts the actions that a user is allowed to perform. It provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

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

Accounting

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

Accounting tracks and maintains a log of every SVS management session. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

AAA Server Groups

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

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

Prerequisites for AAA

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

Guidelines and Limitations

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

AAA Default Settings

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

Configuring AAA

Configuring a Login Authentication Method

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

Before You Begin

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

Procedure

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

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

Enabling Login Authentication Failure Messages

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

The following is the Login Authentication Failure message:

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

Before You Begin

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

Procedure

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

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

Verifying the AAA Configuration

Use one of the following commands to verify the configuration:

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

Command	Purpose
show startup-config aaa	Displays the AAA configuration in the startup configuration.

Example: show aaa authentication

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

Example: show running config aaa

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

Example: show startup-config aaa

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

Configuration Examples for AAA

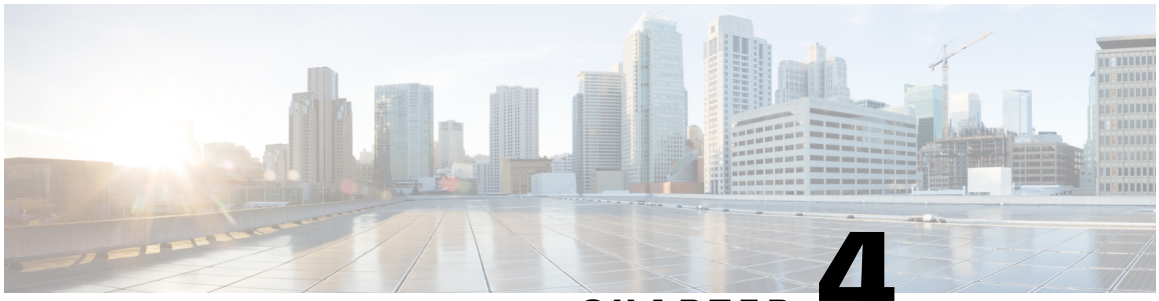
The following is an AAA configuration example:

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

Feature History for AAA

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

Feature Name	Releases	Feature Information
AAA	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 4

Configuring RADIUS

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

Information About RADIUS

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

RADIUS Network Environments

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

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

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.

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

RADIUS Operation

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

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

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

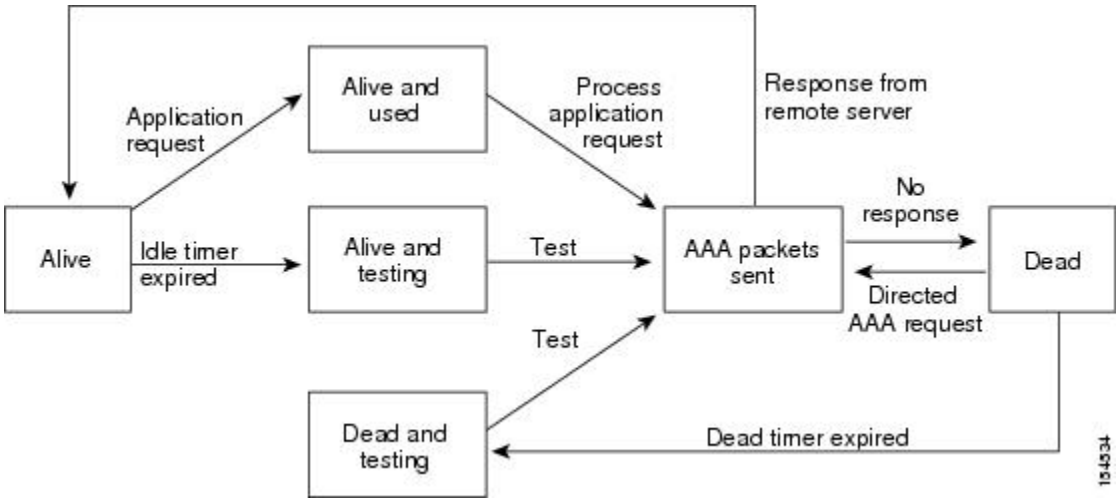
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. Unresponsive RADIUS servers are marked as dead and are not sent AAA requests. Dead RADIUS servers are periodically monitored and returned to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and an error message is displayed indicating that a failure is taking place.

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

Figure 2: Radius Server States



Vendor-Specific Attributes

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

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

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

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

The following VSA protocol options are supported:

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

The following attributes are supported:

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

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

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

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

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

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

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

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

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

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

Prerequisites for RADIUS

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

Guidelines and Limitations

You can configure a maximum of 64 RADIUS servers.

Default Settings

Table 2: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting

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

Configuring RADIUS Servers

Configuring RADIUS Server Hosts

Use this procedure to configure the IP address or the hostname for each RADIUS server to be used for authentication. You should know the following information:

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

Before You Begin

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

Procedure

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

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

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

Configuring the Global RADIUS Key

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

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

Before You Begin

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

Procedure

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

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

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

Configuring a RADIUS Server Key

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

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **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.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the show running-config command to display the encrypted preshared keys.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Configuring RADIUS Server Groups

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

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

Before You Begin

- Before beginning this procedure, you must be logged in to the CLI in EXEC mode.
- All servers in a RADIUS server group must belong to the RADIUS protocol.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration mode for that group. The group-name argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.
Step 3	switch(config-radius)# **server** {*ipv4-address* | *server-name*}	Configures the RADIUS server as a member of the RADIUS server group. **Tip** If the specified RADIUS server is not found, configure it using the radius-server host command and retry this command.
Step 4	switch(config-radius)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	switch(config-radius)# **use-vrf** *vrf-name*	(Optional) Specifies the VRF to use to contact the servers in the server group
Step 6	switch(config-radius)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) Specifies a source interface to be used to reach the RADIUS server. The interface types and interface numbers are defines as follows: • loopback = Virtual interface number from 0 to 1023 • mgmt = Management interface 0 • null = Null interface 0 • port-channel = Port channel number from 1 to 4096
Step 7	switch(config-radius)# **show radius-server groups** [*group-name*]	(Optional) Displays the RADIUS server group configuration.
Step 8	switch(config-radius)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
switch# configure terminal
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# deadtime 30
```

```
switch(config-radius)# use-vrf vrf1
switch(config-radius)# source-interface mgmt0
switch(config-radius)# show radius-server group
total number of groups:2

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

Enabling RADIUS Server Directed Requests

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

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

Directed requests are disabled by default.

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server directed-request**	Enables directed requests. The default is disabled.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch(config)# **show radius-server directed-request**	(Optional) Displays the directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Setting the Global Timeout for All RADIUS Servers

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

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

Before You Begin

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

Procedure

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

```
switch# configure terminal
switch(config)# n1000v(config)# radius-server timeout 101
switch(config-radius)# exit
switch(config-radius)# show radius-server
switch(config-radius)# copy running-config startup-config
```

Configuring a Global Retry Count for All RADIUS Servers

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

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

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

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

Before You Begin

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

Procedure

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

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

Setting the Timeout Interval for a Single RADIUS Server

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

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

Before You Begin

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

Procedure

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

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

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

Configuring Retries for a Single RADIUS Server

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

Before You Begin

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

You should know the following:

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. The default is the global value. **Note** This retransmit count for a single RADIUS server overrides the global setting for all RADIUS servers.
Step 3	switch(config)# **exit**	Returns you to the EXEC mode.
Step 4	switch# **show radius-server**	(Optional) Displays the RADIUS server configuration

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

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

Configuring a RADIUS Accounting Server

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

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

Before You Begin

Before beginning this procedure:

- You must be logged in to the CLI in EXEC mode.
- You should know the destination UDP port number for RADIUS accounting messages.

Procedure

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

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

Configuring a RADIUS Authentication Server

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

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

Before You Begin

Before beginning this procedure:

- You must be logged in to the CLI in EXEC mode.
- You should know the destination UDP port number for RADIUS authentication messages.

Procedure

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

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

Configuring Periodic RADIUS Server Monitoring

Use this procedure to configure the monitoring of RADIUS servers.

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

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

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

Before You Begin

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

Procedure

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

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

Configuring the Global Dead-Time Interval

Use this procedure to configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time to wait after declaring a RADIUS server dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes

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

Before You Begin

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

Procedure

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

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

Manually Monitoring RADIUS Servers or Groups

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

Before You Begin

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

Procedure

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

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

Verifying the RADIUS Configuration

Use one of the following commands to verify the configuration.

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

Displaying RADIUS Server Statistics

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

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

Configuration Example for RADIUS

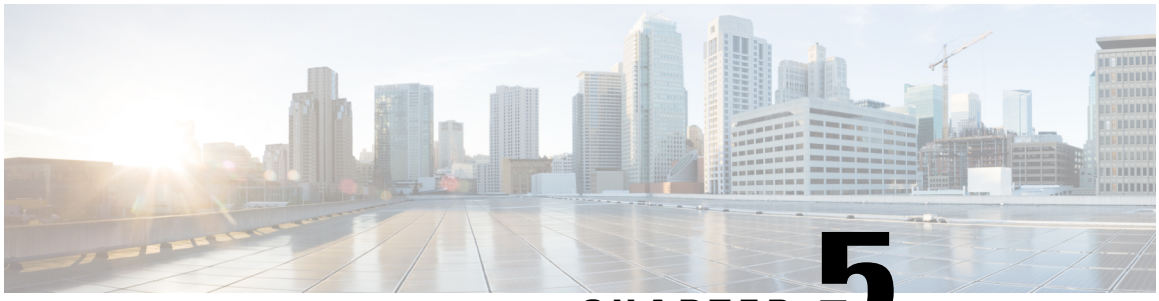
This example shows how to configure a global RADIUS key and a RADIUS server host key:

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

Feature History for RADIUS

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

Feature Name	Releases	Feature Information
RADIUS	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 5

Configuring TACACS+

This chapter contains the following sections:

Information About TACACS+

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

TACACS+ provides for separate authentication, authorization, and accounting services. The TACACS+ daemon provides each service independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. Centralized authentication is provided using the TACACS+ protocol.

TACACS+ Operation for User Login

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

1 When a connection is established, the TACACS+ daemon is contacted to obtain the username and password.

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

2 The TACACS+ daemon provides one of the following responses:

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

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

   c ERROR—An error occurred at some time during authentication either at the daemon or in the network connection. If an ERROR response is received, the device tries to use an alternative method for authenticating the user.

   If further authorization is required after authentication, the user also undergoes an additional authorization phase. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

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

   Services include the following:

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

Default TACACS+ Server Encryption Type and Preshared Key

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

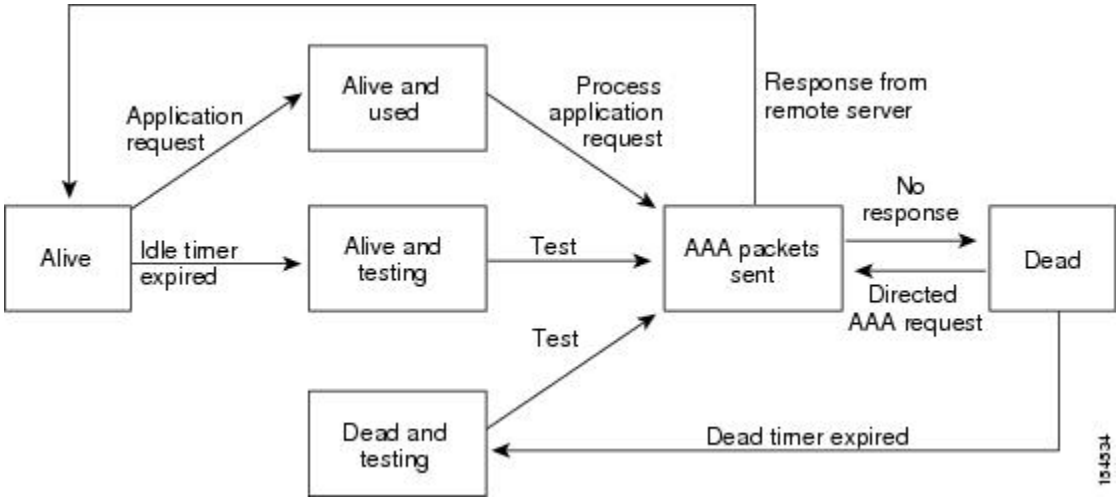
You can override the global preshared key assignment by explicitly using the key option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

Unresponsive TACACS+ servers are marked as dead and are not sent AAA requests. Dead TACACS+ servers are periodically monitored and brought back alive once they respond. This process confirms that a TACACS+ server is in a working state before real AAA requests are sent its way. The following figure shows how a

TACACS+ server state change generates a Simple Network Management Protocol (SNMP) trap and an error message showing the failure before it impacts performance.

Figure 3: TACACS+ Server States



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

Vendor-Specific Attributes

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

Cisco VSA Format

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

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

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

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

The following VSA protocol options are supported:

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

The following attributes are other supported:

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

Prerequisites for TACACS+

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

Guidelines and Limitations for TACACS+

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

Default Settings for TACACS+

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

Configuring TACACS+

The following flowchart steps you through configuring TACACS+

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

Figure 4: Configuring TACACS+ Flowchart

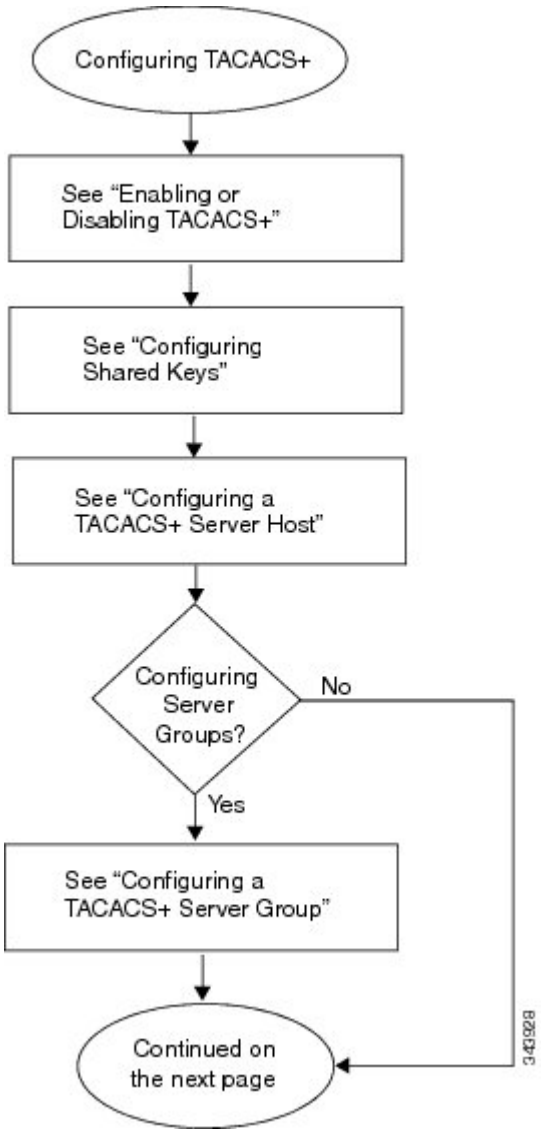


Figure 5: Configuring TACACS+ Flowchart (continued)

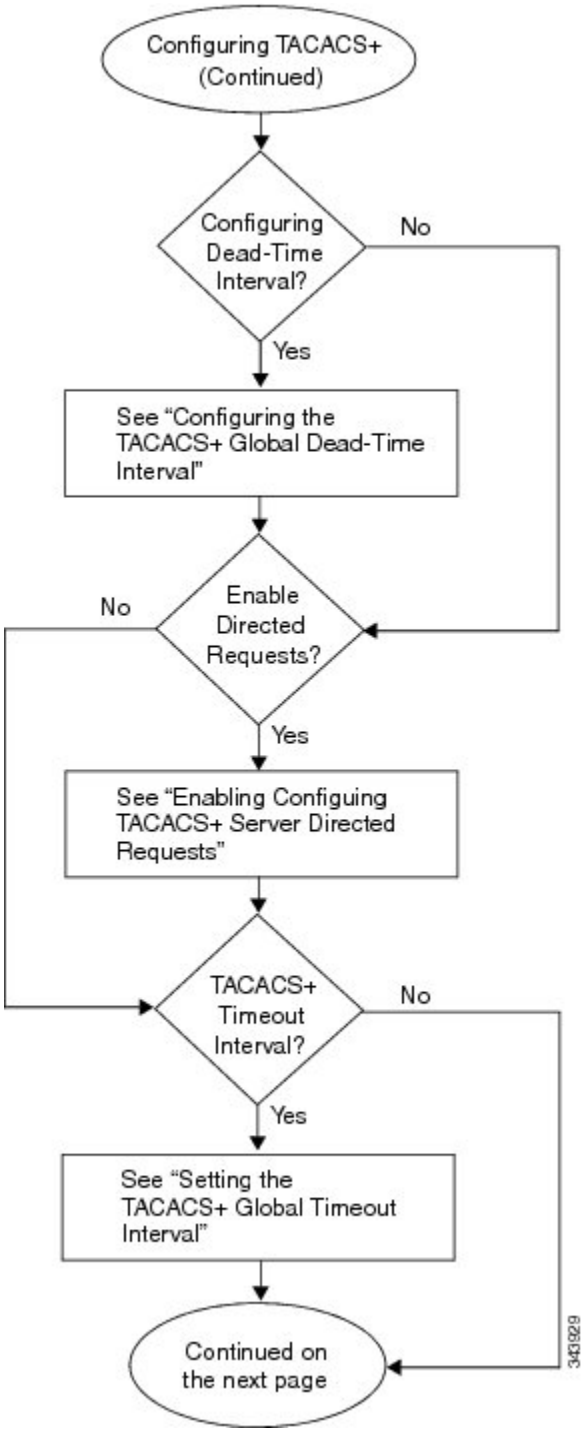
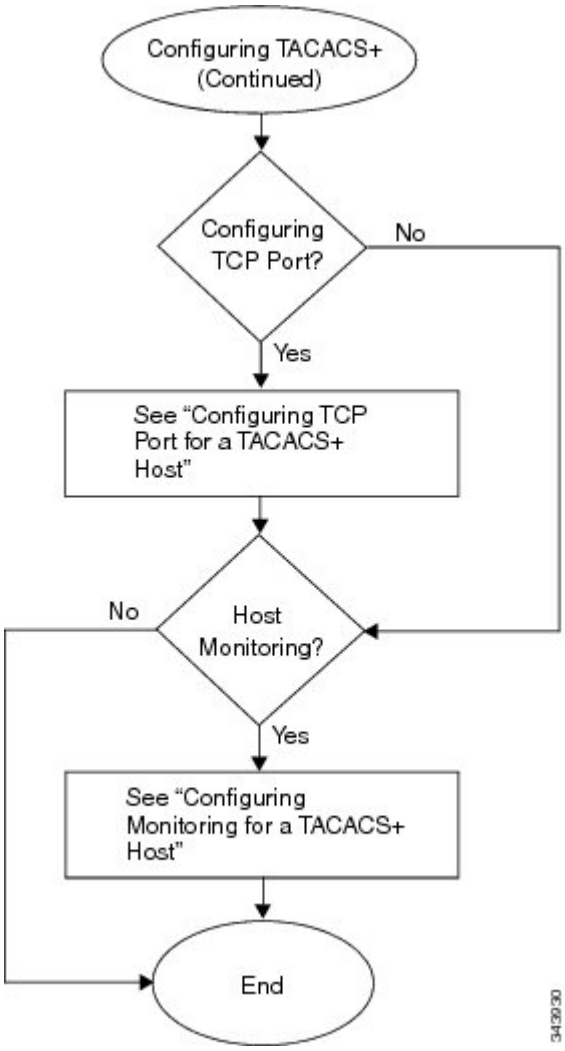


Figure 6: Configuring TACACS+ Flowchart (continued)



Enabling or Disabling TACACS+

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

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# [**no**] **tacacs+ enable**	Enables or disables TACACS+.
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

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

Configuring Shared Keys

By default, no global key is configured.

Use this procedure to configure the following:

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

Before You Begin

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

Procedure

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

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

```
switch# configure terminal
switch(config)# tacacs-server key 0 QsEFtkI#
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:5
deadtime value:0
total number of servers:1

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

Configuring a TACACS+ Server Host

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

Before You Begin

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

- Logged in to the CLI in EXEC mode.
- Enabled TACACS+ for authentication.
- Configured the shared key.
- Know the IP addresses or the hostnames for the remote TACACS+ server hosts.

Procedure

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

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

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

Configuring a TACACS+ Server Group

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

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

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

Before You Begin

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**t	Places you into global configuration mode.
Step 2	switch(config)# **aaa group server tacacs+** *group-name*	Creates a TACACS+ server group with the specified name and places you into the TACACS+ configuration mode for that group.
Step 3	switch(config-tacacs+)# **server** { *ipv4-address* | *host-name*}	Configures the TACACS+ server hostname or IP address as a member of the TACACS+ server group. If the specified TACACS+ server is not found, configure it using the tacacs-server host command and retry this command. **Note** If the specified TACACS+ server is not found, configure it using the tacacs-server host command and retry this command.
Step 4	switch(config-tacacs+)# **deadtime** *minutes*	(Optional) Configures the monitoring dead time for this TACACS+ group. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	switch(config-tacacs+)# **use-vrf** *vrf-name*	(Optional) Specifies the virtual routing and forwarding instance (VRF) to use to contact this server group
Step 6	switch(config-tacacs+)# **source-interface** {*interface-type*} {*interface-number*}	(Optional) Specifies a source interface to be used to reach the TACACS+ server. • loopback = Virtual interface number from 0 to 1023 • mgmt = Management interface 0 • null = Null interface 0 • port-channel = Port channel number from 1 to 4096

	Command or Action	Purpose
Step 7	switch(config-tacacs+)# **show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration
Step 8	switch(config-tacacs+)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

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

Enabling TACACS+ Server Directed Requests

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

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

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

Before You Begin

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Enables use of directed requests for specifying the TACACS+ server to send an authentication request to when logging in. The default is disabled.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 4	switch(config)# **show tacacs-server directed-request**	(Optional) Displays the TACACS+ directed request configuration.
Step 5	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration

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

Setting the TACACS+ Global Timeout Interval

Use this procedure to set the interval in seconds that the Cisco Nexus 1000V waits for a response from any TACACS+ server before declaring a timeout.

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

Before You Begin

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

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

Procedure

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

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

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

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

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

Setting a Timeout Interval for an Individual TACACS+ Host

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

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

Before You Begin

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

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

Procedure

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

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

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

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

Configuring the TCP Port for a TACACS+ Host

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

Before You Begin

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

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

Procedure

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

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

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

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

Configuring Monitoring for a TACACS+ Host

You should know the following information:

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

Before You Begin

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **tacacs-server host** { *ipv4-address* | *host-name* } **test** { **idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]] }	Configures server monitoring. The keywords and arguments are as follows: • **username**: The default is test. **Note** To protect network security, we recommend assigning a username that is not already in the TACACS+ database.

	Command or Action	Purpose
		• **password**: The default is test. • **idle-time**: The default is 0 minutes. The valid range is from 0 to 1440 minutes **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the duration of time in minutes before checking a TACACS+ server that was previously unresponsive. The default value is 0 minutes and the valid range is from 0 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits the global configuration mode and returns you to EXEC mode.
Step 5	switch(config)# **show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.2.2 test username pvk2 password a3z9yjqz7 idle-time
 3
switch(config)# tacacs-server dead-time 5
switch(config)# exit
switch# show tacacs-server
Global TACACS+ shared secret:********
timeout value:10
deadtime value:5
total number of servers:1

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

Configuring the TACACS+ Global Dead-Time Interval

Use this procedure to configure the interval to wait before sending a test packet to a previously unresponsive server.

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

Before You Begin

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

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

- Configured the TACACS+ server.

Procedure

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

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

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

Displaying Statistics for a TACACS+ Host

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

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

Configuration Example for TACACS+

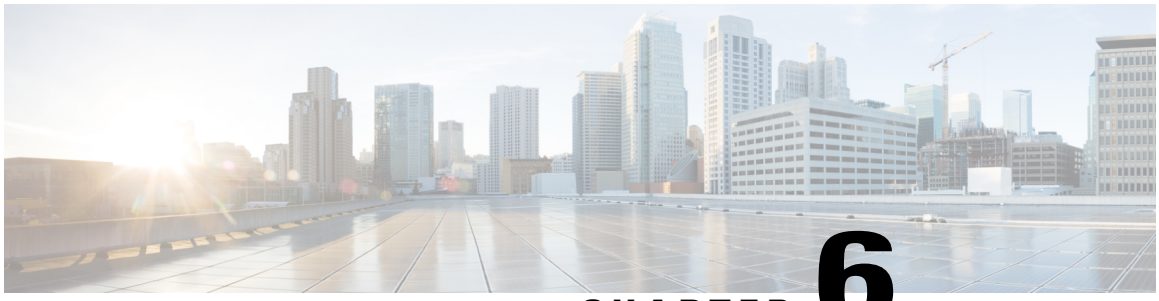
The following example shows a TACACS+ configuration:

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

Feature History for TACACS+

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

Feature Name	Releases	Feature Information
TACACS+	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 6

Configuring SSH

This chapter contains the following sections:

Information about SSH

SSH Server

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

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

SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a secure, encrypted connection to any device that runs the SSH server. This connection provides an encrypted outbound connection. With authentication and encryption, the SSH client produces secure communication over an insecure network.

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

SSH Server Keys

SSH requires server keys for secure communication. You can use SSH server keys for the following SSH options:

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

Be sure to have an SSH server key-pair with the correct version before enabling the SSH service. Generate the SSH server key-pair according to the SSH client version used. The SSH service accepts two types of key-pairs for use by SSH version 2:

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

By default, an RSA key that uses 1024 bits is generated.

SSH supports the following public key formats

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

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

Prerequisites for SSH

SSH has the following prerequisites:

- Configure IP on a Layer 3 interface, out-of-band on the mgmt 0 interface.
- Before enabling the SSH server, obtain the SSH key.

Guidelines and Limitations for SSH

SSH has the following guidelines and limitations

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

Default Settings

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 2048 bits
RSA key bits for generation	1024

Configuring SSH

Generating SSH Server Keys

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

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

Before You Begin

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

Procedure

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

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

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

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

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

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

Configuring a User Account with a Public Key

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

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

Configuring an OpenSSH Key

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

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

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

Before You Begin

Before beginning this procedure, be sure you have:

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

Procedure

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

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

Configuring IETF or PEM Keys

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

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

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

Before You Begin

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

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

Procedure

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

```
switch# copy tftp://10.78.1.10/secsh_file.pub bootflash:secsh_file.pub vrf management
Trying to connect to tftp server......
Connection to server Established.
|
TFTP get operation was successful
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:user2
```

```
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki1OOId9/tdHHa/
ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6
mWoM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+
fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJN
U1JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==
switch# copy running-config startup-config
```

Starting SSH Sessions

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

Before You Begin

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

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

Procedure

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

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

Clearing SSH Hosts

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

Procedure

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

```
switch# clear ssh hosts
```

Disabling the SSH Server

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

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

Before You Begin

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

Procedure

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

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

Deleting SSH Server Keys

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

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server.
Step 3	switch(config)# **no ssh key** [**dsa** | **rsa**]	Deletes the SSH server key.

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

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

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

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

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

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

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

bitcount:1024
fingerprint:
44:91:32:1f:7a:d1:83:3c:f3:5e:db:53:0a:2d:ce:69
**************************************
mcs-srvr43(config)# no ssh key dsa
mcs-srvr43(config)# show ssh key
**************************************
```

```
could not retrieve rsa key information
************************************
could not retrieve dsa key information
************************************
no ssh keys present. you will have to generate them
************************************
```

Clearing SSH Sessions

Use this procedure to clear SSH sessions from the device.

Before You Begin

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

Procedure

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

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

Verifying the SSH Configuration

Use one of the following commands to verify the configuration.

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

Configuration Example for SSH

This example shows the steps you use to configure SSH with an OpenSSH key.

1. Disable the SSH server.

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

2. Generate an SSH server key.

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

3. Enable the SSH server.

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

4. Display the SSH server key.

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

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

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

5. Specify the SSH public key in OpenSSH format.

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

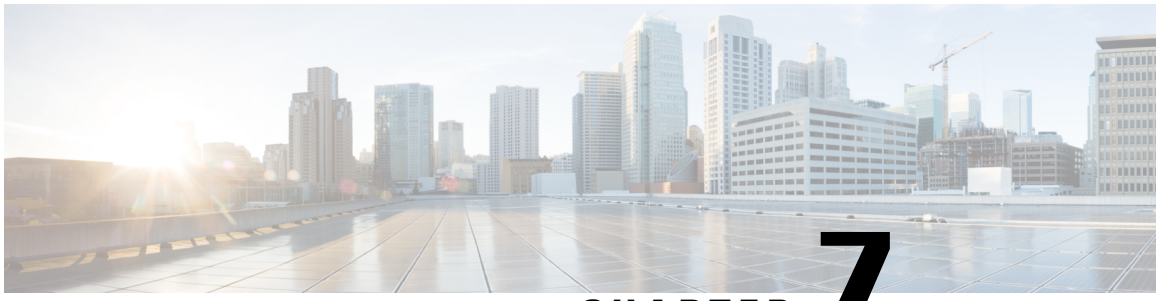
6. Save the configuration.

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

Feature History for SSH

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

Feature Name	Releases	Feature Information
SSH	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 7

Configuring Telnet

This chapter contains the following sections:

Information About the Telnet Server

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

Prerequisites for Telnet

You have configured IP on a Layer 3 interface, out of band on the mgmt 0 interface.

Guidelines and Limitations for Telnet

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

Default Setting for Telnet

Parameter	Default
Telnet server	Disabled

Configuring Telnet

Enabling the Telnet Server

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **feature telnet**	Enables the Telnet server.
Step 3	switch(config)# **show telnet server**	Enables the Telnet server.
Step 4	switch(config)# **show telnet server**	(Optional) Displays the Telnet server configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

Starting an IP Telnet Session to a Remote Device

Before You Begin

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

- Logged in to the CLI in EXEC mode

- Verified that the Telnet server is enabled and it is also enabled on the remote device
- Obtained the hostname for the remote device and, if needed, the username on the remote device

Procedure

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

```
switch# telnet 10.10.1.1
```

Clearing Telnet Sessions

Before You Begin

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

Procedure

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

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

Verifying the Telnet Configuration

Use one of the following commands to verify the configuration.

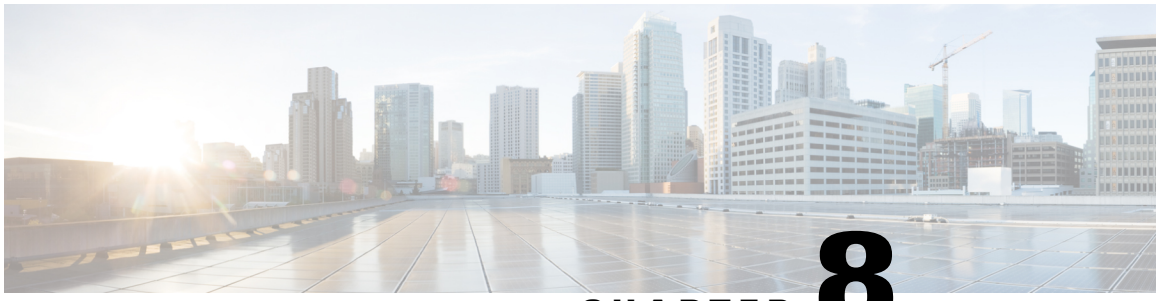
Command	Purpose
show running-config security [**all**]	Displays the user account configuration in the running configuration. The all keyword displays the default values for the user accounts.
show telnet server	Displays the telnet server configuration.
show hosts	Displays the configuration details for current hosts.
show tcp connection	Displays connection information.

```
switch# show running-config security all
version 4.0(1)
username admin password 5 $1$xMw2Q/1S$ZEWRvyAxAJAFV0weuSPvg1  role network-admin
username user2 password 5 $1$byNNnnSP$xfXVKjE5UEScvriwX3Kyj0  role network-operator
username user2 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAyKcb7Nv9Ki100Id9/tdHHa/ngQujlvK5mXyL/n+DeOXKfVhHbX2a+V0cm7CCLUkBh+BvZRmpmOVTmU/5awfVhVxMKXMiPOPBc+A6/n3FVroyRwupMki6mW
oM6UwaGID5gsVPqFjFNSgMWtbhjo97XVKhgjFW+wOVt8QoAcrEtnwEfsnQk1EIr/0XIP1mqTsrqTsmjZ2vLk+fFzTGYAxMvYZI+BrN47aoH2ywS7CpnODjCDXJuDYSPbc3PA8t0ghU/60m9R+s6AZPuljVQbGfxPrahEu4GVc6sMJNU1
JxmqDJkodhMArObB4Umzj7E3Rdby/ZWx/clTYiXQR1X1VfhQ==
telnet server enable

banner motd # User Access Verification #

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

Feature History for Telnet

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

Feature Name		Feature Information
Telnet	Release 5.2(1)IC1(1.1)	This feature was introduced.



CHAPTER 8

Configuring IP ACLs

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

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

Information About ACLs

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

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

ACL Types and Applications

In Cisco Nexus 1000V InterCloud, ACL can be only applied on port profiles .In Cisco Nexus 1000V InterCloud application of ACL on vEthernet Interfaces is not supported.

In Cisco Nexus 1000V InterCloud, IP ACL is supported for traffic filtering. In IP ACL, the device applies IPv4 ACLs only to IP traffic.

Note In this release, MAC ACL is not supported on Cisco Nexus 1000V InterCloud.

Order of ACL Application

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

1. Ingress port ACL
2. Egress port ACL

Rules

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

You can create rules in ACLs in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

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

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

In IP ACLs, you can specify protocols by the integer that represents the Internet protocol number. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

Implicit Rules

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

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

```
deny ip any any
```

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

Additional Filtering Options

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

- IP ACLs support the following additional filtering options:
  - Layer 4 protocol
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP typesVersion1 only
  - Precedence level
  - Differentiated Services Code Point (DSCP) value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set

Sequence Numbers

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

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

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

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

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

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

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

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

Prerequisites for IP ACLs

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

Guidelines and Limitations for IP ACLs

- In most cases, ACL processing for IP packets are processed on the I/O modules. Management interface traffic is always processed on the supervisor module, which is slower.
- IP ACLs can be applied only on port profiles and not on the Interfaces.
- IP ACLs should not be applied on port profiles allowing system vlans.
- If a non-existing ACL is applied on a port profile, a new ACL with the specified name is created and all traffic in the applied port profile is blocked due to the implicit deny.

Default Settings for IP ACLs

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

Configuring IP ACLs

Creating an IP ACL

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# [**no**] **ip access-list** {*name* | **match-local-traffic**}	Creates the named IP ACL (up to 64 characters in length) and enters IP ACL configuration mode.

	Command or Action	Purpose
		The match-local-traffic option enables matching for locally-generated traffic. The no option removes the specified access list.
Step 3	switch(config-acl)# [*sequence-number*] { **permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The sequence-number argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic
Step 4	switch(config-acl)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 5	switch(config-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# show ip access-lists acl-01
switch(config-acl)# copy running-config startup-config
```

Changing an IP ACL

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

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Places you in IP ACL configuration mode for the specified ACL.
Step 3	switch(config-acl)# [*sequence-number*] { **permit** | **deny**} *protocol source destination*	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end

	Command or Action	Purpose
		of the rules. The sequence-number argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic.
Step 4	switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination}*	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** keywords support many ways of identifying traffic.
Step 5	switch(config-acl)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 6	switch(config-acl)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration

```
switch# configure terminal
switch(config)# show ip access-list acl-01
switch(config)# 80 permit ip 192.168.1.0/24 any
switch(config)# ip access-list acl-01
switch(config-acl)# 100 permit ip 192.168.2.0/24 any
switch(config-acl)# no 80
switch(config-acl)# show ip access-lists acl-01
switch(config-acl)# copy running-config startup-config
```

Removing an IP ACL

Removing an ACL does not affect the configuration of the interfaces where applied. Instead, the device considers the removed ACL to be empty and denies all traffic due to the implicit deny rule.

Before You Begin

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **no ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.

	Command or Action	Purpose
Step 3	switch(config)# **show ip access-list** *name* **summary**	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

Changing Sequence Numbers in an IP ACL

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode.
Step 2	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The starting-sequence-number argument and the increment argument can be a whole number from 1 to 4294967295.
Step 3	switch(config)# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 4	switch(config)# **show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# resequence access-list ip acl-01 100 10
```

```
switch(config)# show ip access-lists acl-01
switch(config)# copy running-config startup-config
```

Adding an IP ACL to a Port Profile

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

You must know the following information:

- If you want to create a new port profile, you must know the name you want to give the profile.
- The name of the IP access control list that you want to configure for this port profile.
- The direction of the packet flow for the access list.

Before You Begin

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

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

Procedure

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

```
switch# configure terminal
switch(config)# port-profile AccessProf
switch(config-port-prof)# ip port access-group allaccess4 out
switch(config-port-prof)# show port-profile name AccessProf
switch(config-port-prof)# copy running-config startup-config
```

Applying an IP ACL to the Management Interface

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

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

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Places you into global configuration mode
Step 2	switch(config)# **interface mgmt0**	Places you into interface configuration mode for the management interface.
Step 3	switch(config-if)# [**no**] **ip access-group***access-list* [**in** | **out**]	Applies a specified inbound or outbound IPv4 ACL to the interface. The no option removes the specified configuration.
Step 4	switch(config-if)# **show ip access-lists** *access-list*	(Optional) Displays the ACL configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# interface mgmt0
switch(config-if)# ip access-group telnet in
switch(config-if)# show ip access-lists telnet summary
IP access list telnet
statistics per-entry
Total ACEs Configured:2

Configured on interfaces:
mgmt0 - ingress (Router ACL)

Active on interfaces:
mgmt0 - ingress (Router ACL)
switch(config-if)# copy running-config startup-config
```

Verifying the IP ACL Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including the IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists [*name*]	Displays all IPv4 access control lists (ACLs) or a named IPv4 ACL.
show ip access-list [*name*] su**mmary**	Displays a summary of all configured IPv4 ACLs or a named IPv4 ACL.
show running-config port profile	Displays the configuration of a port profile to which you have applied an ACL.

Monitoring IP ACLs

Use one of the following commands for IP ACL monitoring:

Command	Purpose
show ip access-lists	Displays IPv4 ACL configuration.
show ip access-summary	Displays details about the interfaces that have access lists configured on them.

Feature History for IP ACLs

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

Feature History	Releases	Feature Information
IP ACLs	Release 5.2(1)IC1(1.1)	This feature was introduced.