



System Security Configuration Guide for Cisco NCS 6000 Series Routers, IOS XR Release 6.2.x

First Published: 2017-03-15

Last Modified: 2017-07-14

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2017 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface xi

Changes to This Document xi

Communications, Services, and Additional Information xi

CHAPTER 1

New and Changed Security Features 1

System Security Features Added or Modified in IOS XR Release 6.2.x 1

CHAPTER 2

Configuring AAA Services 3

Prerequisites for Configuring AAA Services 4

Restrictions for Configuring AAA Services 4

Information About Configuring AAA Services 4

User, User Groups, and Task Groups 4

 User Categories 5

 User Groups 6

 Task Groups 6

Command Access in XR and Admin Modes 7

Cisco IOS XR Software Administrative Model 8

 Administrative Access 9

 AAA Database 9

 Remote AAA Configuration 10

 AAA Configuration 10

 Authentication 11

Password Types 12

AAA Password Security for FIPS Compliance 13

 AAA Password Security Policies 13

Task-Based Authorization 15

Task IDs	16
General Usage Guidelines for Task IDs	16
Task IDs for TACACS+ and RADIUS Authenticated Users	17
Task Maps	17
Privilege Level Mapping	19
XML Schema for AAA Services	19
About RADIUS	20
Network Security Situations in Which RADIUS is Unsuitable	21
RADIUS Operation	21
How to Configure AAA Services	22
Configuring Task Groups	22
Task Group Configuration	22
Configuring User Groups	23
Configuring Users	25
Configure AAA Password Policy	27
Configuring Router to RADIUS Server Communication	28
Configuring RADIUS Dead-Server Detection	31
Configuring Per VRF AAA	33
New Vendor-Specific Attributes (VSAs)	33
Configuring a TACACS+ Server	35
Configuring RADIUS Server Groups	38
Configuring TACACS+ Server Groups	40
Configure Per VRF TACACS+ Server Groups	41
Configuring AAA Method Lists	43
Configuring Authentication Method Lists	43
Configuring Authorization Method Lists	45
Configuring Accounting Method Lists	48
Generating Interim Accounting Records	51
Applying Method Lists for Applications	52
Enabling AAA Authorization	52
Enabling Accounting Services	54
Configuring Login Parameters	55
Configuring a TACACS+ Server on the Sysadmin Plane	56
Configuring TACACS+ Server	56

Configuring Authorization for a TACACS+ Server	57
Configuring Authentication for a TACACS+ Server	59
Configuring Accounting for a TACACS+ Server	60
Configuration Examples for Configuring AAA Services	60
Configuring AAA Services: Example	61
Feature Overview	62
Prerequisites for Model Based AAA	62
Initial Operation	62
NACM Configuration Management and Persistence	64
Overview of Configuring NACM	64
NACM Rules	64
Enabling NACM	67
Verify the NACM Configurations	67
Disabling NACM	67
Additional References	68
<hr/>	
CHAPTER 3	Implementing Certification Authority Interoperability 71
Prerequisites for Implementing Certification Authority	71
Restrictions for Implementing Certification Authority	72
Information About Implementing Certification Authority	72
Supported Standards for Certification Authority Interoperability	72
Certification Authorities	73
Purpose of CAs	73
IPSec Without CAs	73
IPSec with CAs	74
IPSec with Multiple Trustpoint CAs	74
How IPSec Devices Use CA Certificates	74
CA Registration Authorities	75
How to Implement CA Interoperability	75
Configuring a Router Hostname and IP Domain Name	75
Generating an RSA Key Pair	76
Importing a Public Key to the Router	77
Declaring a Certification Authority and Configuring a Trusted Point	77
Authenticating the CA	79

Requesting Your Own Certificates 79

Configuring Certificate Enrollment Using Cut-and-Paste 80

Configuration Examples for Implementing Certification Authority Interoperability 82

 Configuring Certification Authority Interoperability: Example 82

Where to Go Next 84

Additional References 84

CHAPTER 4

Implementing Keychain Management 87

Prerequisites for Configuring Keychain Management 87

Restrictions for Implementing Keychain Management 87

Information About Implementing Keychain Management 87

 Lifetime of Key 88

How to Implement Keychain Management 88

 Configuring a Keychain 88

 Configuring a Tolerance Specification to Accept Keys 89

 Configuring a Key Identifier for the Keychain 90

 Configuring the Text for the Key String 91

 Determining the Valid Keys 93

 Configuring the Keys to Generate Authentication Digest for the Outbound Application Traffic 94

 Configuring the Cryptographic Algorithm 95

Configuration Examples for Implementing Keychain Management 96

 Configuring Keychain Management: Example 96

Additional References 97

CHAPTER 5

Implementing Lawful Intercept 99

Prerequisites for Implementing Lawful Intercept 99

Restrictions for Implementing Lawful Intercept 101

Information About Lawful Intercept Implementation 101

 Interception Mode 102

 Overlapping Taps 102

 Provisioning for VoIP Calls 102

 Call Interception 102

 Provisioning for Data Sessions 103

 Data Interception 103

Lawful Intercept Topology	103
Intercepting IPv4 and IPv6 Packets	104
Lawful Intercept Filters	104
Intercepting Packets Based on Flow ID (Applies to IPv6 only)	105
Intercepting VRF (6VPE) and 6PE Packets	105
Encapsulation Type Supported for Intercepted Packets	106
Per Tap Drop Counter Support	106
High Availability for Lawful Intercept	106
Preserving TAP and MD Tables during RP Fail Over	107
Replay Timer	107
Installing Lawful Intercept (LI) Package	107
Installing and Activating the LI Package	107
Deactivating the LI PIE	108
How to Configure SNMPv3 Access for Lawful Intercept on the Cisco ASR 9000 Series Router	109
Disabling SNMP-based Lawful Intercept	109
Configuring the Inband Management Plane Protection Feature	110
Enabling the Mediation Device to Intercept VoIP and Data Sessions	110
Adding MD and TAP Objects	113
Configuration Example for Inband Management Plane Feature Enablement	114
Configuring the Inband Management Plane Protection Feature: Example	114
Additional References	115

CHAPTER 6
Implementing Management Plane Protection 117

Prerequisites for Implementing Management Plane Protection	117
Restrictions for Implementing Management Plane Protection	117
Information About Implementing Management Plane Protection	118
Inband Management Interface	118
Out-of-Band Management Interface	118
Peer-Filtering on Interfaces	119
Control Plane Protection Overview	119
Management Plane	119
Management Plane Protection Feature	119
Benefits of the Management Plane Protection Feature	120
How to Configure a Device for Management Plane Protection	120

Configuring a Device for Management Plane Protection for an Inband Interface 120

Configuring a Device for Management Plane Protection for an Out-of-band Interface 123

Configuration Examples for Implementing Management Plane Protection 125

 Configuring Management Plane Protection: Example 125

Additional References 127

CHAPTER 7 **Configuring Software Authentication Manager 129**

Prerequisites for Configuring Software Authentication Manager 129

Information about Software Authentication Manager 129

How to set up a Prompt Interval for the Software Authentication Manager 130

CHAPTER 8 **Implementing Secure Shell 131**

Prerequisites for Implementing Secure Shell 132

Restrictions for Implementing Secure Shell 132

Information About Implementing Secure Shell 133

 SSH Server 133

 SSH Client 133

 SFTP Feature Overview 135

 RSA Based Host Authentication 136

 RSA Based User Authentication 136

 SSHv2 Client Keyboard-Interactive Authentication 137

The Network Configuration Protocol 137

 Netconf Sessions and Operations 138

 The Yang data model 138

How to Implement Secure Shell 139

 Configuring SSH 139

 Configuring the SSH Client 142

Enabling NETCONF over SSH 144

Configuration Examples for Implementing Secure Shell 145

 Configuring Secure Shell: Example 146

Examples: Netconf over SSH 146

Additional References 147

CHAPTER 9 **Implementing Secure Socket Layer 149**

Prerequisites for Implementing Secure Shell	150
Restrictions for Implementing Secure Shell	150
Information About Implementing Secure Shell	151
SSH Server	151
SSH Client	151
SFTP Feature Overview	153
RSA Based Host Authentication	154
RSA Based User Authentication	154
SSHv2 Client Keyboard-Interactive Authentication	155
The Network Configuration Protocol	155
Netconf Sessions and Operations	156
The Yang data model	156
How to Implement Secure Shell	157
Configuring SSH	157
Configuring the SSH Client	160
Enabling NETCONF over SSH	162
Configuration Examples for Implementing Secure Shell	163
Configuring Secure Shell: Example	164
Examples: Netconf over SSH	164
Additional References	165



Preface



Note This product has reached end-of-life status. For more information, see the [End-of-Life and End-of-Sale Notices](#).

This guide describes the configuration and examples for system security. For system security command descriptions, usage guidelines, task IDs, and examples, refer to the *System Security Command Reference for Cisco NCS 6000 Series Routers*.

The preface contains the following sections:

- [Changes to This Document, on page xi](#)
- [Communications, Services, and Additional Information, on page xi](#)

Changes to This Document

This table lists the technical changes made to this document since it was first released.

Table 1: Changes to This Document

Date	Summary
March 2017	Initial release of this document.
July 2017	Republished for Release 6.2.2.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco Marketplace](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).

- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

Cisco Bug Search Tool

[Cisco Bug Search Tool](#) (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed Security Features

- [System Security Features Added or Modified in IOS XR Release 6.2.x, on page 1](#)

System Security Features Added or Modified in IOS XR Release 6.2.x

Feature	Description	Changed in Release	Where Documented
None	No new features introduced	Release 6.2.2	Not Applicable
AAA Password Security Policy for FIPS Compliance	This feature was introduced.	Release 6.2.1	<i>Configuring AAA Services</i> chapter: AAA Password Security for FIPS Compliance, on page 13 See the <i>Authentication, Authorization, and Accounting Commands</i> chapter in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i> , for information on the commands related to this feature.



CHAPTER 2

Configuring AAA Services

This module describes the implementation of the administrative model of *task-based authorization* used to control user access in the Cisco IOS XR software system. The major tasks required to implement task-based authorization involve configuring user groups and task groups.

User groups and task groups are configured through the Cisco IOS XR software command set used for authentication, authorization and accounting (AAA) services. Authentication commands are used to verify the identity of a user or principal. Authorization commands are used to verify that an authenticated user (or principal) is granted permission to perform a specific task. Accounting commands are used for logging of sessions and to create an audit trail by recording certain user- or system-generated actions.

AAA is part of the Cisco IOS XR software base package and is available by default.



Note For a complete description of the AAA commands listed in this module, see the *Authentication, Authorization, and Accounting Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Configuring AAA Services

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Configuring AAA Services, on page 4](#)
- [Restrictions for Configuring AAA Services, on page 4](#)
- [Information About Configuring AAA Services, on page 4](#)
- [How to Configure AAA Services, on page 22](#)
- [Configuring a TACACS+ Server on the Sysadmin Plane, on page 56](#)
- [Configuration Examples for Configuring AAA Services, on page 60](#)
- [Feature Overview, on page 62](#)
- [Overview of Configuring NACM, on page 64](#)
- [Disabling NACM, on page 67](#)
- [Additional References, on page 68](#)

Prerequisites for Configuring AAA Services

The following are the prerequisites to configure AAA services:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Establish a root system user using the initial setup dialog. The administrator may configure a few local users without any specific AAA configuration. The external security server becomes necessary when user accounts are shared among many routers within an administrative domain. A typical configuration would include the use of an external AAA security server and database with the local database option as a backup in case the external server becomes unreachable.

Restrictions for Configuring AAA Services

This section lists the restrictions for configuring AAA services.

Compatibility

Compatibility is verified with the Cisco freeware TACACS+ server and FreeRADIUS only.

Interoperability

Router administrators can use the same AAA server software and database (for example, CiscoSecure ACS) for the router and any other Cisco equipment that does not currently run Cisco IOS XR software. To support interoperability between the router and external TACACS+ servers that do not support task IDs, see the “[Task IDs for TACACS+ and RADIUS Authenticated Users, on page 17](#)” section.

Information About Configuring AAA Services

This section lists all the conceptual information that a Cisco IOS XR software user must understand before configuring user groups and task groups through AAA or configuring Remote Authentication Dial-in User Service (RADIUS) or TACACS+ servers. Conceptual information also describes what AAA is and why it is important.

User, User Groups, and Task Groups

Cisco IOS XR software user attributes form the basis of the Cisco IOS XR software administrative model. Each router user is associated with the following attributes:

- User ID (ASCII string) that identifies the user uniquely across an administrative domain
- Length limitation of 253 characters for passwords and one-way encrypted secrets
- List of user groups (at least one) of which the user is a member (thereby enabling attributes such as task IDs). (See the [Task IDs, on page 16](#) section)

User Categories

Router users are classified into the following categories:

- Root Secure Domain Router (SDR) user (specific SDR administrative authority)
- SDR user (specific SDR user access)

Root System Users

The root system user is the entity authorized to “own” the entire router chassis. The root system user functions with the highest privileges over all router components and can monitor all secure domain routers in the system. At least one root system user account must be created during router setup. Multiple root system users can exist.

The root system user can perform any configuration or monitoring task, including the following:

- Configure secure domain routers.
- Create, delete, and modify root SDR users (after logging in to the secure domain router as the root system user). (See the [Root SDR Users, on page 5](#) section.)
- Create, delete, and modify secure domain router users and set user task permissions (after logging in to the secure domain router as the root system user). (See the [Secure Domain Router \(SDR\) Users, on page 5](#) section.)
- Access fabric racks or any router resource not allocated to a secure domain router, allowing the root system user to authenticate to any router node regardless of the secure domain router configurations.

Root SDR Users

A root SDR user controls the configuration and monitoring of a particular SDR. The root SDR user can create users and configure their privileges within the SDR. Multiple root SDR users can work independently. A single SDR may have more than one root SDR user.

A root SDR user can perform the following administrative tasks for a particular SDR:

- Create, delete, and modify secure domain router users and their privileges for the SDR. (See the [Secure Domain Router \(SDR\) Users, on page 5](#) section.)
- Create, delete, and modify user groups to allow access to the SDR.
- Manage nearly all aspects of the SDR.

A root SDR user cannot deny access to a root system user. (See the [Root System Users, on page 5](#) section.)

Secure Domain Router (SDR) Users

A SDR user has restricted access to an SDR as determined by the root SDR user. The SDR user performs the day-to-day system and network management activities. The tasks that the secure domain router user is allowed to perform are determined by the task IDs associated with the user groups to which the SDR user belongs. (See the [User Groups, on page 6](#) section.)



Note Multiple SDRs in a chassis are not supported.

User Groups

A *user group* defines a collection of users that share a set of attributes, such as access privileges. Cisco IOS XR software allows the system administrator to configure groups of users and the job characteristics that are common in groups of users. Users are not assigned to groups by default hence the assignment needs to be done explicitly. A user can be assigned to more than one group.

Each user may be associated with one or more user groups. User groups have the following attributes:

- A user group consists of the list of task groups that define the authorization for the users. All tasks, except `cisco-support`, are permitted by default for root system users. (See the [Root System Users, on page 5](#) section.)
- Each user task can be assigned read, write, execute, or debug permission.

Predefined User Groups

The Cisco IOS XR software provides a collection of user groups whose attributes are already defined. The predefined groups are as follows:

- **cisco-support:** This group is used by the Cisco support team.
- **netadmin:** Has the ability to control and monitor all system and network parameters.
- **operator:** A demonstration group with basic privileges.
- **root-1r:** Has the ability to control and monitor the specific secure domain router.
- **sysadmin:** Has the ability to control and monitor all system parameters but cannot configure network protocols.
- **serviceadmin:** Service administration tasks, for example, Session Border Controller (SBC).

User-Defined User Groups

Administrators can configure their own user groups to meet particular needs.

User Group Inheritance

A user group can derive attributes from another user group. (Similarly, a task group can derive attributes from another task group). For example, when user group A inherits attributes from user group B, the new set of task attributes of the user group A is a union of A and B. The inheritance relationship among user groups is dynamic in the sense that if group A inherits attributes from group B, a change in group B affects group A, even if the group is not reinherited explicitly.

Task Groups

A task group is defined by a collection of task IDs. Task groups contain task ID lists for each class of action.

Each user group is associated with a set of task groups applicable to the users in that group. A user's task permissions are derived from the task groups associated with the user groups to which that user belongs.

Predefined Task Groups

The following predefined task groups are available for administrators to use, typically for initial configuration:

- **cisco-support:** Cisco support personnel tasks

- **netadmin:** Network administrator tasks
- **operator:** Operator day-to-day tasks (for demonstration purposes)
- **root-lr:** Secure domain router administrator tasks
- **sysadmin:** System administrator tasks
- **serviceadmin:** Service administration tasks, for example, SBC

User-Defined Task Groups

Users can configure their own task groups to meet particular needs.

Group Inheritance

Task groups support inheritance from other task groups. (Similarly, a user group can derive attributes from another user group. See the [User Groups, on page 6](#) section.) For example, when task group A inherits task group B, the new set of attributes of task group A is the union of A and B.

Command Access in XR and Admin Modes

The XR user group and task is mapped to the System Admin VM group when the System Admin mode is accessed from XR mode using **admin** command. The corresponding access permission on System Admin VM is available to the user. Currently, only aaa, admin task and root-lr groups are mapped to System Admin VM group or task. The other tasks like protocols are not mapped as these services are not supported in System Admin VM. The disaster-recovery user of System Admin VM is synced with the Host VM.

XR Task or Group	Sysadmin VM Group	Access	Example
root-lr	Root-system group	Full access to the system configuration.	<pre>RP/0/RP0/CPU0:ios#show user group Mon Nov 3 13:48:54.536 UTC root-lr, cisco-support RP/0/RP0/CPU0:ios#show user tasks inc root-lr Mon Nov 3 13:49:06.495 UTC Task: root-lr : READ WRITE EXECUTE DEBUG (reserved) RP/0/RP0/CPU0:ios#admin sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:48:00.790 UTC User group : root-system</pre>

XR Task or Group	Sysadmin VM Group	Access	Example
Admin-r/w/x/d	Admin-r	Read only commands on Sysadmin VM	<pre> taskgroup tg-admin-write task write admin task execute admin ! usergroup ug-admin-write taskgroup tg-admin-write ! username admin-write group ug-admin-write password admin-write ! RP/0/RP0/CPU0:ios#show user group Mon Nov 3 14:09:29.676 UTC ug-admin-write RP/0/RP0/CPU0:ios#show user tasks Mon Nov 3 14:09:35.244 UTC Task: admin : READ WRITE EXECUTE RP/0/RP0/CPU0:ios#admin Mon Nov 3 14:09:40.401 UTC admin-write connected from 127.0.0.1 using console on xr-vm_node0_RP0_CPU0 sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:53:00.790 UTC User group : admin-r </pre>
Netadmin or sysadmin group Admin-r/ wx/d, aaa -r/w/x/d	Aaa -r and admin -r	Read only commands on Sysadmin VM	<pre> RP/0/RP0/CPU0:ios#show user group Mon Nov 3 13:44:39.176 UTC netadmin RP/0/RP0/CPU0:ios#show user tasks inc aaa Mon Nov 3 13:45:00.999 UTC Task: aaa : READ RP/0/RP0/CPU0:ios#show user tasks inc admin Mon Nov 3 13:45:09.567 UTC Task: admin : READ RP/0/RP0/CPU0:ios#admin Mon Nov 3 13:46:21.183 UTC netadmin connected from 127.0.0.1 using console on xr-vm_node0_RP0_CPU0 sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:44:23.939 UTC User group : admin-r,aaa-r sysadmin-vm:0_RP0# </pre>

Cisco IOS XR Software Administrative Model

The router operates in two planes: the administration (admin) plane and secure domain router (SDR) plane. The admin (shared) plane consists of resources shared across all SDRs, while the SDR plane consists of those resources specific to the particular SDR.

Each SDR has its own AAA configuration including, local users, groups, and TACACS+ and RADIUS configurations. Users created in one SDR cannot access other SDRs unless those same users are configured in the other SDRs.

Administrative Access

Administrative access to the system can be lost if the following operations are not well understood and carefully planned.

- Configuring authentication that uses remote AAA servers that are not available, particularly authentication for the console.



Note The **none** option without any other method list is not supported in Cisco IOS XR software.

- Configuring command authorization or XR EXEC mode authorization on the console should be done with extreme care, because TACACS+ servers may not be available or may deny every command, which locks the user out. This lockout can occur particularly if the authentication was done with a user not known to the TACACS+ server, or if the TACACS+ user has most or all the commands denied for one reason or another.

To avoid a lockout, we recommend one or both of the following:

- Before turning on TACACS+ command authorization or XR EXEC mode authorization on the console, make sure that the user who is configuring the authorization is logged in using the appropriate user permissions in the TACACS+ profile.
- If the security policy of the site permits it, use the **none** option for command authorization or XR EXEC mode authorization so that if the TACACS+ servers are not reachable, AAA rolls over to the **none** method, which permits the user to run the command.

AAA Database

The AAA database stores the users, groups, and task information that controls access to the system. The AAA database can be either local or remote. The database that is used for a specific situation depends on the AAA configuration.

Local Database

AAA data, such as users, user groups, and task groups, can be stored locally within a secure domain router. The data is stored in the in-memory database and persists in the configuration file. The stored passwords are encrypted.



Note The database is local to the specific secure domain router (SDR) in which it is stored, and the defined users or groups are not visible to other SDRs in the same system.

You can delete the last remaining user from the local database. If all users are deleted when the next user logs in, the setup dialog appears and prompts you for a new username and password.



Note The setup dialog appears only when the user logs into the console.

Remote Database

AAA data can be stored in an external security server, such as CiscoSecure ACS. Security data stored in the server can be used by any client (such as a network access server [NAS]) provided that the client knows the server IP address and shared secret.

Remote AAA Configuration

Products such as CiscoSecure ACS can be used to administer the shared or external AAA database. The router communicates with the remote AAA server using a standard IP-based security protocol (such as TACACS+ or RADIUS).

Client Configuration

The security server should be configured with the secret key shared with the router and the IP addresses of the clients.

User Groups

User groups that are created in an external server are not related to the user group concept that is used in the context of local AAA database configuration on the router. The management of external TACACS+ server or RADIUS server user groups is independent, and the router does not recognize the user group structure. The remote user or group profiles may contain attributes that specify the groups (defined on the router) to which a user or users belong, as well as individual task IDs. For more information, see the [Task IDs for TACACS+ and RADIUS Authenticated Users, on page 17](#) section.

Configuration of user groups in external servers comes under the design of individual server products. See the appropriate server product documentation.

Task Groups

Task groups are defined by lists of permitted task IDs for each type of action (such as read, write, and so on). The task IDs are basically defined in the router system. Task ID definitions may have to be supported before task groups in external software can be configured.

Task IDs can also be configured in external TACACS+ or RADIUS servers.

AAA Configuration

This section provides information about AAA configuration.

Method Lists

AAA data may be stored in a variety of data sources. AAA configuration uses *method lists* to define an order of preference for the source of AAA data. AAA may define more than one method list and applications (such as login) can choose one of them. For example, console ports may use one method list and the vty ports may use another. If a method list is not specified, the application tries to use a default method list. If a default method list does not exist, AAA uses the local database as the source.

Rollover Mechanism

AAA can be configured to use a prioritized list of database options. If the system is unable to use a database, it automatically rolls over to the next database on the list. If the authentication, authorization, or accounting request is rejected by any database, the rollover does not occur and the request is rejected.

The following methods are available:

- Local: Use the locally configured database (not applicable for accounting and certain types of authorization)
- TACACS+: Use a TACACS+ server (such as CiscoSecure ACS)
- RADIUS: Use a RADIUS server
- Line: Use a line password and user group (applicable only for authentication)
- None: Allow the request (not applicable for authentication)

Server Grouping

Instead of maintaining a single global list of servers, the user can form server groups for different AAA protocols (such as RADIUS and TACACS+) and associate them with AAA applications (such as PPP and XR EXEC mode).

Authentication

Authentication is the most important security process by which a principal (a user or an application) obtains access to the system. The principal is identified by a username (or user ID) that is unique across an administrative domain. The applications serving the user (such as or Management Agent) procure the username and the credentials from the user. AAA performs the authentication based on the username and credentials passed to it by the applications. The role of an authenticated user is determined by the group (or groups) to which the user belongs. (A user can be a member of one or more user groups.)

Authentication of Non-Owner Secure Domain Router User

When logging in from a non-owner secure domain router, the root system user must add the “@admin” suffix to the username. Using the “@admin” suffix sends the authentication request to the owner secure domain router for verification. The owner secure domain router uses the methods in the list-name **remote** for choosing the authentication method. The **remote** method list is configured using the **aaa authentication login remote method1 method2...** command. (See the [Configuring AAA Method Lists, on page 43](#) section.)

Authentication of Owner Secure Domain Router User

An owner secure domain router user can log in only to the nodes belonging to the specific secure domain router associated with that owner secure domain router user. If the user is member of a root-sdr group, the user is authenticated as an owner secure domain router user.

Authentication of Secure Domain Router User

Secure domain router user authentication is similar to owner secure domain router user authentication. If the user is not found to be a member of the designated owner secure domain router user group, the user is authenticated as a secure domain router user.

Authentication Flow of Control

AAA performs authentication according to the following process:

1. A user requests authentication by providing a username and password (or secret).
2. AAA verifies the user’s password and rejects the user if the password does not match what is in the database.
3. AAA determines the role of the user (root SDR user, or SDR user).

- If the user has been configured as a member of an owner secure domain router user group, then AAA authenticates the user as an owner secure domain router user.
- If the user has not been configured as a member of an owner secure domain router user group, AAA authenticates the user as a secure domain router user.

Clients can obtain a user's permitted task IDs during authentication. This information is obtained by forming a union of all task group definitions specified in the user groups to which the user belongs. Clients using such information typically create a session for the user (such as an API session) in which the task ID set remains static. Both the XR EXEC mode and external API clients can use this feature to optimize their operations. XR EXEC mode can avoid displaying the commands that are not applicable and an EMS application can, for example, disable graphical user interface (GUI) menus that are not applicable.

If the attributes of a user, such as user group membership and, consequently, task permissions, are modified, those modified attributes are not reflected in the user's current active session; they take effect in the user's next session.

Authentication Failure

In a system which is configured either with TACACS+ or RADIUS authentication with AAA configuration similar to the configuration below during the first login attempt or attempts, following a system reload, the login to the RP auxiliary port fails.

```
aaa authentication login default group tacacs+ group radius local
line template aux
login authentication default
```

This is because following the reload, the auxiliary port rejects login attempts with a valid TACACS+ configured *username* and *password*.

In such a scenario, the user has to first login with a valid locally configured *username* and *password*, and any login thereafter with TACACS+ configured *username* and *password*. Alternatively, if the user is connected to the auxiliary port via a terminal server, first clear the line used on the terminal server itself, and thereafter the user will be able to login to the auxiliary port with the TACACS+ configured *username* and *password*.

Password Types

In configuring a user and that user's group membership, you can specify two types of passwords: encrypted or clear text.

The router supports both two-way and one-way (secret) encrypted user passwords. Secret passwords are ideal for user login accounts because the original unencrypted password string cannot be deduced on the basis of the encrypted secret. Some applications (PPP, for example) require only two-way passwords because they must decrypt the stored password for their own function, such as sending the password in a packet. For a login user, both types of passwords may be configured, but a warning message is displayed if one type of password is configured while the other is already present.

If both secret and password are configured for a user, the secret takes precedence for all operations that do not require a password that can be decrypted, such as login. For applications such as PPP, the two-way encrypted password is used even if a secret is present.

AAA Password Security for FIPS Compliance

Cisco IOS XR Software introduces advanced AAA password strengthening policy and security mechanism to store, retrieve and provide rules or policy to specify user passwords. This password policy is applicable only for local users, and not for remote users whose profile information are stored in a third party AAA server. This policy is not applicable to secrets of the user. If both secret and password are configured for a user, then secret takes precedence, and password security policy does not have any effect on authentication or change of password for such users. This AAA password security policy works as such for Cisco IOS XR platforms. Whereas, this feature is supported only on XR VM, for Cisco IOS XR 64 bit platforms and Cisco NCS 6000 Series Routers.

High Availability for AAA Password Security Policy

The AAA password policy configurations and username configurations remain intact across RP failovers or process restarts in the system. The operational data such as, lifetime of the password and lockout time of the user are not stored on system database or disk. Hence, those are not restored across RP failovers or process restarts. Users start afresh on the active RP or on the new process. Hence, users who were locked out before RP failover or process restart are able to login immediately after the failover or restart.

To configure AAA password policy, see [Configure AAA Password Policy, on page 27](#).

AAA Password Security Policies

AAA password security for FIPS compliance consists of these policies:

Password Composition Policy

Passwords can be composed by any combination of upper and lower case alphabets, numbers and special characters that include: "!", "@", "#", "\$", "%", "^", "&", "*", "(", and ")". Security administrator can also set the types and number of required characters that comprise the password, thereby providing more flexibility for password composition rules. The minimum number of character change required between passwords is 4, by default. There is no restriction on the upper limit of the number of uppercase, lowercase, numeric and special characters.

Password Length Policy

The administrator can set the minimum and maximum length of the password. The minimum configurable length in password policy is 2, and the maximum length is 253.

Password Lifetime Policy

The administrator can configure a maximum lifetime for the password, the value of which can be specified in years, months, days, hours, minutes and seconds. The configured password never expires if this parameter is not configured. The configuration remains intact even after a system reload. But, the password creation time is updated to the new time whenever the system reboots. For example, if a password is configured with a life time of one month, and if the system reboots on 29th day, then the password is valid for one more month after the system reboot. Once the configured lifetime expires, further action is taken based on the password expiry policy (see the section on Password Expiry Policy).

Password Expiry Policy

If the password credential of a user who is trying to login is already expired, then the following actions occur:

- User is prompted to set the new password after successfully entering the expired password.

- The new password is validated against the password security policy.
- If the new password matches the password security policy, then the AAA data base is updated and authentication is done with the new password.
- If the new password is not compliant with the password security policy, then the attempt is considered as an authentication failure and the user is prompted again to enter a new password. The max limit for such attempts is in the control of login clients and AAA does not have any restrictions for that.

As part of password expiry policy, if the life time is not yet configured for a user who has already logged in, and if the security administrator configures the life time for the same user, then the life time is set in the database. The system checks for password expiry on the subsequent authentication of the same user.

Password expiry is checked only during the authentication phase. If the password expires after the user is authenticated and logged in to the system, then no action is taken. The user is prompted to change the password only during the next authentication of the same user.

Debug logs and syslog are printed for the user password expiry only when the user attempts to login. This is a sample syslog in the case of password expiry:

```
RP/0/RSP1/CPU0:Jun 21 09:13:34.241 : locald_DSC[308]: %SECURITY-LOCALD-5-USER_PASSWD_EXPIRED
:
Password for user 'user12' has expired.
```

Password Change Policy

Users cannot change passwords at will. A password change is triggered in these scenarios:

- When the security administrator needs to change the password
- When the user is trying to get authenticated using a profile and the password for the profile is expired
- When the security administrator modifies the password policy which is associated to the user, and does not immediately change the password according to the policy

You can use the **show configuration failed** command to display the error messages when the password entered does not comply with the password policy configurations.

When the security administrator changes the password security policy, and if the existing profile does not meet the password security policy rules, no action is taken if the user has already logged in to the system. In this scenario, the user is prompted to change the password when he tries to get authenticated using the profile which does not meet the password security rules.

When the user is changing the password, the lifetime of the new password remains same as that of the lifetime that was set by the security administrator for the old profile.

When password expires for non-interactive clients (such as dot1x), an appropriate error message is sent to the clients. Clients must contact the security administrator to renew the password in such scenarios.

Service Provision after Authentication

The basic AAA local authentication feature ensures that no service is performed before a user is authenticated.

User Re-authentication Policy

A user is re-authenticated when he changes the password. When a user changes his password on expiry, he is authenticated with the new password. In this case, the actual authentication happens based on the previous credential, and the new password is updated in the database.

User Authentication Lockout Policy

AAA provides a configuration option, **authen-max-attempts**, to restrict users who try to authenticate using invalid login credentials. This option sets the maximum number of permissible authentication failure attempts for a user. The user gets locked out when he exceeds this maximum limit, until the lockout timer (**lockout-time**) is expired. If the user attempts to login in spite of being locked out, the lockout expiry time keep advancing forward from the time login was last attempted.

This is a sample syslog when user is locked out:

```
RP/0/RSP1/CPU0:Jun 21 09:21:28.226 : locald_DSC[308]: %SECURITY-LOCALD-5-USER_PASSWD_LOCKED
:
User 'user12' is temporarily locked out for exceeding maximum unsuccessful logins.
```

This is a sample syslog when user is unlocked for authentication:

```
RP/0/RSP1/CPU0:Jun 21 09:14:24.633 : locald_DSC[308]: %SECURITY-LOCALD-5-USER_PASSWD_UNLOCKED
:
User 'user12' is unlocked for authentications.
```

Password Policy Creation, Modification and Deletion

Security administrators having write permission for AAA tasks are allowed to create password policy. Modification is allowed at any point of time, even when the policy is associated to a user. Deletion of password policy is not allowed until the policy is un-configured from the user.

After the modification of password policy associated with a user, security administrator can decide if he wants to change passwords of associated users complying to the password policy. Based on this, there are two scenarios:

- If the administrator configures the password, then the user is not prompted to change the password on next login.
- If the administrator does not configure the password, then the user is prompted to change the password on next login.

In either of the above cases, at every password expiry interval, the user is prompted to change the password on next login.

Debug messages are printed when password policies are created, modified and deleted.

Task-Based Authorization

AAA employs “task permissions” for any control, configure, or monitor operation through CLI or API. The Cisco IOS software concept of privilege levels has been replaced in Cisco IOS XR software by a task-based authorization system.

Task IDs

The operational tasks that enable users to control, configure, and monitor Cisco IOS XR software are represented by task IDs. A task ID defines the permission to run an operation for a command. Users are associated with sets of task IDs that define the breadth of their authorized access to the router.

Task IDs are assigned to users through the following means:

Each user is associated with one or more user groups. Every user group is associated with one or more *task groups*; in turn, every task group is defined by a set of task IDs. Consequently, a user's association with a particular user group links that user to a particular set of task IDs. A user that is associated with a task ID can execute any operation associated with that task ID.

General Usage Guidelines for Task IDs

Most router control, configuration, or monitoring operation (CLI or XML API) is associated with a particular set of task IDs. Typically, a given CLI command or API invocation is associated with at least one or more task IDs. Neither the **config** nor the **commit** commands require any specific task id permissions. The configuration and commit operations do not require specific task ID permissions. Aliases also don't require any task ID permissions. You cannot perform a configuration replace unless root-lr permissions are assigned. If you want to deny getting into configuration mode you can use the TACACS+ command authorization to deny the config command. These associations are hard-coded within the router and may not be modified. Task IDs grant permission to perform certain tasks; task IDs do not deny permission to perform tasks. Task ID operations can be one, all, or a combination of classes that are listed in this table.

Table 2: Task ID Classes

Operation	Description
Read	Specifies a designation that permits only a read operation.
Write	Specifies a designation that permits a change operation and implicitly allows a read operation.
Execute	Specifies a designation that permits an access operation; for example ping and Telnet.
Debug	Specifies a designation that permits a debug operation.

The system verifies that each CLI command and API invocation conforms with the task ID permission list for the user. If you are experiencing problems using a CLI command, contact your system administrator.

Multiple task ID operations separated by a slash (for example read/write) mean that both operations are applied to the specified task ID.

Multiple task ID operations separated by a comma (for example read/write, execute) mean that both operations are applied to the respective task IDs. For example, the **copy ipv4 access-list** command can have the read and write operations applied to the *acl* task ID, and the execute operation applied to the *filesystem* task ID.

If the task ID and operations columns have no value specified, the command is used without any previous association to a task ID and operation. In addition, users do not have to be associated to task IDs to use ROM monitor commands.

Users may need to be associated to additional task IDs to use a command if the command is used in a specific configuration submode. For example, to execute the **show redundancy** command, a user needs to be associated to the system (read) task ID and operations as shown in the following example:

```
RP/0/RP0/CPU0:router# show redundancy
```

Task IDs for TACACS+ and RADIUS Authenticated Users

Cisco IOS XR software AAA provides the following means of assigning task permissions for users authenticated with the TACACS+ and RADIUS methods:

- Specify the text version of the task map directly in the configuration file of the external TACACS+ and RADIUS servers.
See the “[Task Maps, on page 17](#)” section for more details.
- Specify the privilege level in the configuration file of the external TACACS+ and RADIUS servers.
See the “[Privilege Level Mapping, on page 19](#)” section for more details.
- Create a local user with the same username as the user authenticating with the TACACS+ and RADIUS methods.
- Specify, by configuration, a default task group whose permissions are applied to any user authenticating with the TACACS+ and RADIUS methods.

Task Maps

For users who are authenticated using an external TACACS+ server and RADIUS server, Cisco IOS XR software AAA supports a method to define task IDs remotely.

Format of the Task String

The task string in the configuration file of the TACACS+ server consists of tokens delimited by a comma (.). Each token contains either a task ID name and its permissions or the user group to include for this particular user, as shown in the following example:

```
task = “ permissions : taskid name , # usergroup name , ...”
```



Note Cisco IOS XR software allows you to specify task IDs as an attribute in the external RADIUS or TACACS+ server. If the server is also shared by non-Cisco IOS XR software systems, these attributes are marked as optional as indicated by the server documentation. For example, CiscoSecure ACS and the freeware TACACS+ server from Cisco require an asterisk (*) instead of an equal sign (=) before the attribute value for optional attributes. If you want to configure attributes as optional, refer to the TACACS+ server documentation.

For example, to give a user named user1 BGP read, write, and execute permissions and include user1 in a user group named operator, the username entry in the external server’s TACACS+ configuration file would look similar to the following:

```
user = user1{
member = some-tac-server-group
opap = cleartext "lab"
service = exec {
task = "rwx:bgp,#operator"
}
}
```

The r,w,x, and d correspond to read, write, execute and debug, respectively, and the pound sign (#) indicates that a user group follows.



Note The optional keyword must be added in front of “task” to enable interoperability with systems based on Cisco IOS software.

If CiscoSecure ACS is used, perform the following procedure to specify the task ID and user groups:

SUMMARY STEPS

1. Enter your username and password.
2. Click the **Group Setup** button to display the **Group Setup** window.
3. From the Group drop-down list, select the group that you want to update.
4. Click the **Edit Settings** button.
5. Use the scroll arrow to locate the Shell (exec) check box.
6. Check the **Shell (exec)** check box to enable the custom attributes configuration.
7. Check the **Custom attributes** check box.
8. Enter the following task string without any blank spaces or quotation marks in the field:
9. Click the **Submit + Restart** button to restart the server.

DETAILED STEPS

- Step 1** Enter your username and password.
- Step 2** Click the **Group Setup** button to display the **Group Setup** window.
- Step 3** From the Group drop-down list, select the group that you want to update.
- Step 4** Click the **Edit Settings** button.
- Step 5** Use the scroll arrow to locate the Shell (exec) check box.
- Step 6** Check the **Shell (exec)** check box to enable the custom attributes configuration.
- Step 7** Check the **Custom attributes** check box.
- Step 8** Enter the following task string without any blank spaces or quotation marks in the field:

Example:

```
task=rwx:bgp,#netadmin
```

- Step 9** Click the **Submit + Restart** button to restart the server.

The following RADIUS Vendor-Specific Attribute (VSA) example shows that the user is part of the sysadmin predefined task group, can configure BGP, and can view the configuration for OSPF:

Example:

```
user Auth-Type := Local, User-Password == lab
  Service-Type = NAS-Prompt-User,
  Reply-Message = "Hello, %u",
  Login-Service = Telnet,
  Cisco-AVPair = "shell:tasks=#sysadmin,rwx:bgp,r:ospf"
```

After user1 successfully connects and logs in to the external TACACS+ server with username user1 and appropriate password, the **show user tasks** command can be used in XR EXEC mode to display all the tasks user1 can perform. For example:

Example:

```

Username:user1
Password:
RP/0/RP0/CPU0:router# show user tasks

Task:      basic-services  :READ   WRITE   EXECUTEDEBUG
Task:      bgp             :READ   WRITE   EXECUTE
Task:      cdp             :READ
Task:      diag            :READ
Task:      ext-access      :READ           EXECUTE
Task:      logging         :READ

```

Alternatively, if a user named user2, who does not have a task string, logs in to the external server, the following information is displayed:

Example:

```

Username:user2
Password:
RP/0/RP0/CPU0:router# show user tasks
No task ids available

```

Privilege Level Mapping

For compatibility with TACACS+ daemons that do not support the concept of task IDs, AAA supports a mapping between privilege levels defined for the user in the external TACACS+ server configuration file and local user groups. Following TACACS+ authentication, the task map of the user group that has been mapped from the privilege level returned from the external TACACS+ server is assigned to the user. For example, if a privilege level of 5 is returned from the external TACACS server, AAA attempts to get the task map of the local user group priv5. This mapping process is similar for other privilege levels from 1 to 13. For privilege level 14 maps to the user group owner-sdr.

For example, with the Cisco freeware tac plus server, the configuration file has to specify *priv_lvl* in its configuration file, as shown in the following example:

```

user = sampleuser1{
  member = bar
  service = exec-ext {
    priv_lvl = 5
  }
}

```

The number 5 in this example can be replaced with any privilege level that has to be assigned to the user *sampleuser*.

XML Schema for AAA Services

The extensible markup language (XML) interface uses requests and responses in XML document format to configure and monitor AAA. The AAA components publish the XML schema corresponding to the content and structure of the data used for configuration and monitoring. The XML tools and applications use the schema to communicate to the XML agent for performing the configuration.

The following schema are published by AAA:

- Authentication, Authorization and Accounting configuration
- User, user group, and task group configuration
- TACACS+ server and server group configuration
- RADIUS server and server group configuration

About RADIUS

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

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its AAA security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, and local username lookup.



Note

RADIUS is supported on all Cisco platforms, but some RADIUS-supported features run only on specified platforms.

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

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a "smart card" access control system. In one case, RADIUS has been used with Enigma security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco router with RADIUS to the network. This might be the first step when you make a transition to a Terminal Access Controller Access Control System Plus (TACACS+) server.
- Networks in which a user must access only a single service. Using RADIUS, you can control user access to a single host, utility such as Telnet, or protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 10.2.3.4 and the defined access list is started.
- 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 RADIUS access control and accounting software to meet special security and billing needs.

- Networks that support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions and to efficiently manage the use of shared resources to offer differing service-level agreements.

Network Security Situations in Which RADIUS is Unsuitable

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
 - AppleTalk Remote Access (ARA)
 - NetBIOS Frame Control Protocol (NBFCP)
 - NetWare Asynchronous Services Interface (NASI)
 - X.25 PAD connections
- Router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a router other than a Cisco router if that router requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

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:
 - a. ACCEPT—The user is authenticated.
 - a. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
 - a. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
 - a. 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 used for XR EXEC mode 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 PPP, Serial Line Internet Protocol (SLIP), or XR EXEC mode services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

How to Configure AAA Services

To configure AAA services, perform the tasks described in the following sections.

Configuring Task Groups

Task-based authorization employs the concept of a *task ID* as its basic element. A task ID defines the permission to execute an operation for a given user. Each user is associated with a set of permitted router operation tasks identified by task IDs. Users are granted authority by being assigned to user groups that are in turn associated with task groups. Each task group is associated with one or more task IDs. The first configuration task in setting up an authorization scheme to configure the task groups, followed by user groups, followed by individual users.

Task Group Configuration

Task groups are configured with a set of task IDs per action type.

Specific task IDs can be removed from a task group by specifying the **no** prefix for the **task** command.

The task group itself can be removed. Deleting a task group that is still referred to elsewhere results in an error.

Before you begin

Before creating task groups and associating them with task IDs, you should have some familiarity with the router list of task IDs and the purpose of each task ID. Use the **show aaa task supported** command to display a complete list of task IDs.



Note Only users with write permissions for the AAA task ID can configure task groups.

SUMMARY STEPS

1. **configure**
2. **taskgroup** *taskgroup-name*
3. **description** *string*
4. **task** {**read** | **write** | **execute** | **debug**} *taskid-name*
5. Repeat for each task ID to be associated with the task group named in Step 2.
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters XR Config mode.

	Command or Action	Purpose
Step 2	taskgroup <i>taskgroup-name</i> Example: RP/0/RP0/CPU0:router(config)# taskgroup beta	Creates a name for a particular task group and enters task group configuration submenu. <ul style="list-style-type: none"> Specific task groups can be removed from the system by specifying the no form of the taskgroup command.
Step 3	description <i>string</i> Example: RP/0/RP0/CPU0:router(config-tg)# description this is a sample task group description	(Optional) Creates a description of the task group named in Step 2.
Step 4	task { read write execute debug } <i>taskid-name</i> Example: RP/0/RP0/CPU0:router(config-tg)# task read bgp	Specifies a task ID to be associated with the task group named in Step 2. <ul style="list-style-type: none"> Assigns read permission for any CLI or API invocations associated with that task ID and performed by a member of the task group. Specific task IDs can be removed from a task group by specifying the no prefix for the task command.
Step 5	Repeat for each task ID to be associated with the task group named in Step 2.	—
Step 6	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session. No —Exits the configuration session without committing the configuration changes. Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After completing configuration of a full set of task groups, configure a full set of user groups as described in the Configuring User Groups section.

Configuring User Groups

User groups are configured with the command parameters for a set of users, such as task groups. Entering the **usergroup** command accesses the user group configuration submenu. Users can remove specific user groups by using the **no** form of the **usergroup** command. Deleting a usergroup that is still referenced in the system results in a warning.

Before you begin

Note Only users associated with the WRITE:AAA task ID can configure user groups. User groups cannot inherit properties from predefined groups, such as owner-sdr.

SUMMARY STEPS

1. **configure**
2. **usergroup** *usergroup-name*
3. **description** *string*
4. **inherit usergroup** *usergroup-name*
5. **taskgroup** *taskgroup-name*
6. Repeat Step for each task group to be associated with the user group named in Step [Step 2, on page 24](#).
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	usergroup <i>usergroup-name</i> Example: RP/0/RP0/CPU0:router(config)# usergroup beta	Creates a name for a particular user group and enters user group configuration submenu. <ul style="list-style-type: none"> • Specific user groups can be removed from the system by specifying the no form of the usergroup command.
Step 3	description <i>string</i> Example: RP/0/RP0/CPU0:router(config-ug)# description this is a sample user group description	(Optional) Creates a description of the user group named in Step Step 2, on page 24 .
Step 4	inherit usergroup <i>usergroup-name</i> Example: RP/0/RP0/CPU0:router(config-ug)# inherit usergroup sales	<ul style="list-style-type: none"> • Circular inclusions are detected and rejected. Permissions may not be inherited from predefined user groups. • To explicitly define permissions for the user group named in Step Step 2, on page 24, omit Step Step 4, on page 24 and go to Step 5, on page 24.
Step 5	taskgroup <i>taskgroup-name</i> Example:	Associates the user group named in Step Step 2, on page 26 with the task group named in this step.

	Command or Action	Purpose
	<code>RP/0/RP0/CPU0:router(config-ug)# taskgroup beta</code>	<ul style="list-style-type: none"> The user group takes on the configuration attributes (task ID list and permissions) already defined for the entered task group.
Step 6	Repeat Step for each task group to be associated with the user group named in Step Step 2, on page 24 .	—
Step 7	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session. No —Exits the configuration session without committing the configuration changes. Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After completing configuration of a full set of user groups, configure individual users as described in the [Configuring Users, on page 25](#) section.

Configuring Users

Perform this task to configure a user.

Each user is identified by a username that is unique across the administrative domain. Each user should be made a member of at least one user group. Deleting a user group may orphan the users associated with that group. The AAA server authenticates orphaned users but most commands are not authorized.

SUMMARY STEPS

- configure**
- username** *user-name*
- Do one of the following:
 - password** [**0** | **7**] *password*
 - .
- group** *group-name*
- Repeat [Step 4, on page 26](#) for each user group to be associated with the user specified in [Step 2, on page 26](#).
- Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	username <i>user-name</i> Example: RP/0/RP0/CPU0:router(config)# username user1	Creates a name for a new user (or identifies a current user) and enters username configuration submenu. <ul style="list-style-type: none"> The <i>user-name</i> argument can be only one word. Spaces and quotation marks are not allowed.
Step 3	Do one of the following: <ul style="list-style-type: none"> password [0 7] <i>password</i> . Example: RP/0/RP0/CPU0:router(config-un)# password 0 pwd1 or RP/0/RP0/CPU0:router(config-un)# secret 0 sec1	Specifies a password for the user named in Step 2, on page 26 . <ul style="list-style-type: none"> Use the secret command to create a secure login password for the user names specified in Step 2, on page 26. Entering 0 following the password command specifies that an unencrypted (clear-text) password follows. Entering 7 following the password command specifies that an encrypted password follows. . Type 0 is the default for the password and secret commands.
Step 4	group <i>group-name</i> Example: RP/0/RP0/CPU0:router(config-un)# group sysadmin	Assigns the user named in Step 2, on page 26 to a user group that has already been defined through the usergroup command. <ul style="list-style-type: none"> The user takes on all attributes of the user group, as defined by that user group's association to various task groups. Each user must be assigned to at least one user group. A user may belong to multiple user groups.
Step 5	Repeat Step 4, on page 26 for each user group to be associated with the user specified in Step 2, on page 26 .	—
Step 6	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After completing configuration of a full set of users, configure router to use the RADIUS server communication or TACACS+ servers (See the [Configuring Router to RADIUS Server Communication, on page 28](#) or [Configuring a TACACS+ Server, on page 35](#) section.)

Configure AAA Password Policy

To configure the AAA password policy, use the **aaa password-policy** command in the global configuration mode.

Configuration Example

This example shows how to configure a AAA password security policy, *test-policy*. This *test-policy* is applied to a user by using the **username** command along with **password-policy** option.

```
RP/0/RP0/CPU0:router(config)#aaa password-policy test-policy
RP/0/RP0/CPU0:router(config-aaa)#min-length 8
RP/0/RP0/CPU0:router(config-aaa)#max-length 15
RP/0/RP0/CPU0:router(config-aaa)#lifetime months 3
RP/0/RP0/CPU0:router(config-aaa)#min-char-change 5
RP/0/RP0/CPU0:router(config-aaa)#authen-max-attempts 3
RP/0/RP0/CPU0:router(config-aaa)#lockout-time days 1
RP/0/RP0/CPU0:router(config-aaa)#commit

RP/0/RP0/CPU0:router(config)#username user1 password-policy test-policy password 0 pwd1
```

Running Configuration

```
aaa password-policy test-policy
  min-length 8
  max-length 15
  lifetime months 3
  min-char-change 5
  authen-max-attempts 3
  lockout-time days 1
!
```

Verification

Use this command to get details of the AAA password policy configured in the router:

```
RP/0/RP0/CPU0:router#show aaa password-policy
```

```
Fri Feb 3 16:50:58.086 EDT
Password Policy Name : test-policy
Number of Users : 1
Minimum Length : 8
Maximum Length : 15
Special Character Len : 0
Uppercase Character Len : 0
Lowercase Character Len : 1
Numeric Character Len : 0
Policy Life Time :
  seconds : 0
  minutes : 0
  hours : 0
  days : 0
  months : 3
  years : 0
Lockout Time :
  seconds : 0
  minutes : 0
  hours : 0
  days : 1
  months : 0
  years : 0
Character Change Len : 5
Maximum Failure Attempts : 3
```

Related Topic

- [AAA Password Security for FIPS Compliance, on page 13](#)

Associated Commands

- `aaa password-policy`
- `show aaa password-policy`
- `username`

Configuring Router to RADIUS Server Communication

This task configures router to RADIUS server communication.

The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (CiscoSecure ACS), Livingston, Merit, Microsoft, or another software provider. Configuring router to RADIUS server communication can have several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Retransmission value
- Timeout period
- Key string

RADIUS security servers are identified on the basis of their hostname or IP address, hostname and specific User Datagram Protocol (UDP) port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port numbers creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as an automatic switchover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order they are configured.)

A RADIUS server and a Cisco router use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the router.

The timeout, retransmission, and encryption key values are configurable globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the router, use the three unique global commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command.



Note You can configure both global and per-server timeout, retransmission, and key value commands simultaneously on the same Cisco network access server. If both global and per-server functions are configured on a router, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

SUMMARY STEPS

1. **configure**
2. **radius-server host** {hostname | ip address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]
3. **radius-server retransmit** retries
4. **radius-server timeout** seconds
5. **radius-server key** {0 clear-text-key | 7 encrypted-key | clear-text-key}
6. **radius source-interface** type instance [vrf vrf-id]
7. Repeat [Step 2, on page 30](#) through [Step 6, on page 31](#) for each external server to be configured.
8. Use the **commit** or **end** command.
9. show radius

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.

	Command or Action	Purpose
Step 2	<p>radius-server host {<i>hostname</i> <i>ip address</i>} [auth-port <i>port-number</i>] [acct-port <i>port-number</i>] [timeout <i>seconds</i>] [retransmit <i>retries</i>] [key <i>string</i>]</p> <p>Example: Specifying a radius server hostname</p> <pre>RP/0//CPU0:router(config)# radius-server host host1</pre> <p>Example: Specifying a radius server in IPv6 format</p> <pre>RP/0//CPU0:router(config)# radius-server host 2001:db8:a0b:12f0::1/64</pre>	<p>Specifies the hostname or IP address of the remote RADIUS server host. The IP address can be specified in IPv4 or IPv6 format.</p> <ul style="list-style-type: none"> Use the auth-port <i>port-number</i> option to configure a specific UDP port on this RADIUS server to be used solely for authentication. Use the acct-port <i>port-number</i> option to configure a specific UDP port on this RADIUS server to be used solely for accounting. To configure the network access server to recognize more than one host entry associated with a single IP address, simply repeat this command as many times as necessary, making sure that each UDP port number is different. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host. If no timeout is set, the global value is used; otherwise, enter a value in the range 1 to 1000. If no retransmit value is set, the global value is used; otherwise enter a value in the range 1 to 100. If no key string is specified, the global value is used. <p>Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius-server host command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.</p>
Step 3	<p>radius-server retransmit <i>retries</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# radius-server retransmit 5</pre>	<p>Specifies the number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up.</p> <ul style="list-style-type: none"> In the example, the number of retransmission attempts is set to 5.
Step 4	<p>radius-server timeout <i>seconds</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# radius-server timeout 10</pre>	<p>Sets the number of seconds a router waits for a server host to reply before timing out.</p> <ul style="list-style-type: none"> In the example, the interval timer is set to 10 seconds.
Step 5	<p>radius-server key {0 <i>clear-text-key</i> 7 <i>encrypted-key</i> <i>clear-text-key</i>}</p> <p>Example:</p>	<p>Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.</p>

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router(config)# radius-server key 0 samplekey</pre>	
Step 6	<p>radius source-interface <i>type instance</i> [vrf <i>vrf-id</i>]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# radius source-interface 0/3/0/1</pre>	<p>(Optional) Forces RADIUS to use the IP address of a specified interface or subinterface for all outgoing RADIUS packets.</p> <ul style="list-style-type: none"> The specified interface or subinterface must have an IP address associated with it. If the specified interface or subinterface does not have an IP address or is in the down state, then RADIUS reverts to the default. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state. <p>The vrf keyword enables the specification on a per-VRF basis.</p>
Step 7	Repeat Step 2, on page 30 through Step 6, on page 31 for each external server to be configured.	—
Step 8	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session. No —Exits the configuration session without committing the configuration changes. Cancel —Remains in the configuration session, without committing the configuration changes.
Step 9	<p>show radius</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show radius</pre>	(Optional) Displays information about the RADIUS servers that are configured in the system.

What to do next

After configuring router to RADIUS server communication, configure RADIUS server groups. (See the [Configuring RADIUS Server Groups, on page 38](#) section.)

Configuring RADIUS Dead-Server Detection

This task configures the RADIUS Dead-Server Detection feature.

The RADIUS Dead-Server Detection feature lets you configure and determine the criteria that is used to mark a RADIUS server as dead. If no criteria is explicitly configured, the criteria is computed dynamically on the basis of the number of outstanding transactions. The RADIUS dead-server detection configuration results in the prompt detection of RADIUS servers that have stopped responding. The prompt detection of nonresponding

RADIUS servers and the avoidance of swamped and dead-to-live-to-dead-again servers result in less deadtime and quicker packet processing.

You can configure the minimum amount of time, in seconds, that must elapse from the time that the router last received a valid packet from the RADIUS server to the time the server is marked as dead. If a packet has not been received since the router booted, and there is a timeout, the time criterion is treated as though it was met.

In addition, you can configure the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. If the server performs both authentication and accounting, both types of packets are included in the number. Improperly constructed packets are counted as though they are timeouts. Only retransmissions are counted, not the initial transmission. For example, each timeout causes one retransmission to be sent.



Note Both the time criterion and the tries criterion must be met for the server to be marked as dead.

The **radius-server deadtime** command specifies the time, in minutes, for which a server is marked as dead, remains dead, and, after this period, is marked alive even when no responses were received from it. When the dead criteria are configured, the servers are not monitored unless the **radius-server deadtime** command is configured.

SUMMARY STEPS

1. **configure**
2. **radius-server deadtime** *minutes*
3. **radius-server dead-criteria time** *seconds*
4. **radius-server dead-criteria tries** *tries*
5. Use the **commit** or **end** command.
6. **show radius dead-criteria host** *ip address in IPv4 or IPv6 format* [**auth-port** *auth-port*] [**acct-port** *acct-port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters XR Config mode.
Step 2	radius-server deadtime <i>minutes</i> Example: RP/0/RP0/CPU0:router(config)# <code>radius-server deadtime 5</code>	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 3	radius-server dead-criteria time <i>seconds</i> Example:	Establishes the time for the dead-criteria conditions for a RADIUS server to be marked as dead.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config)# radius-server dead-criteria time 5	
Step 4	radius-server dead-criteria tries tries Example: RP/0/RP0/CPU0:router(config)# radius-server dead-criteria tries 4	Establishes the number of tries for the dead-criteria conditions for a RADIUS server to be marked as dead.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 6	show radius dead-criteria host ip address in IPv4 or IPv6 format [auth-port auth-port] [acct-port acct-port] Example: RP/0/RP0/CPU0:router# show radius dead-criteria host 172.19.192.80	(Optional) Displays dead-server-detection information that has been requested for a RADIUS server at the specified IP address.

Configuring Per VRF AAA

The Per VRF AAA functionality enables AAA services to be based on VPN routing and forwarding (VRF) instances. The Provider Edge (PE) or Virtual Home Gateway (VHG) communicates directly with the customer's RADIUS server, which is associated with the customer's VPN, without having to go through a RADIUS proxy. Thus, ISPs can scale their VPN offerings more efficiently, because they no longer have to use RADIUS proxies and they can provide their customers with the flexibility they demand.

New Vendor-Specific Attributes (VSAs)

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor-specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco IOS XR software RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair" The value is a string of the following format:

```
protocol : attribute sep value *
```

“Protocol” is a value of the Cisco “protocol” attribute for a particular type of authorization. “Attribute” and “value” are an appropriate attribute-value (AV) pair defined in the Cisco RADIUS specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes.

This table describes the VSAs that are now supported for Per VRF AAA.

Table 3: Supported VSAs for Per VRF AAA

VSA Name	Value Type	Description
Note	The RADIUS VSAs—rad-serv, rad-serv-source-if, and rad-serv-vrf—must have the prefix “aaa:” before the VSA name.	
rad-serv	string	<p>Indicates the IP address in IPv4 or IPv6 format, key, timeout, and retransmit number of a server and the group of the server.</p> <p>The VSA syntax follows:</p> <pre>rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W].</pre> <p>Other than the IP address, all parameters are optional and are issued in any order. If the optional parameters are not specified, their default values are used.</p> <p>The key cannot contain any spaces; for “retransmit V,” “V” can range from 1 to 100; for “timeout W,” the “W” can range from 1 to 1000.</p>
rad-serv-vrf	string	Specifies the name of the VRF that is used to transmit RADIUS packets. The VRF name matches the name that was specified through the vrf command.

This task configures RADIUS server groups per VRF. For information about configuring TACACS+ server groups per VRF, refer [Configuring TACACS+ Server Groups, on page 40](#).

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *group-name*
3. **server-private** {*hostname* | *ip-address in IPv4 or IPv6 format*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
4. **vrf** *vrf-name*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.

	Command or Action	Purpose
Step 2	aaa group server radius <i>group-name</i> Example: <pre>RP/0//CPU0:router(config)# aaa group server radius radgroup1 RP/0//CPU0:router (config-sg-radius) #</pre>	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	server-private { <i>hostname</i> <i>ip-address in IPv4 or IPv6 format</i> } [auth-port <i>port-number</i>] [acct-port <i>port-number</i>] [timeout <i>seconds</i>] [retransmit <i>retries</i>] [key string] Example: IP address in IPv4 format <pre>RP/0//CPU0:router(config-sg-radius) # server-private 10.1.1.1 timeout 5 RP/0//CPU0:router (config-sg-radius) # server-private 10.2.2.2 retransmit 3</pre> Example: IP address in IPv6 format <pre>RP/0//CPU0:router (config-sg-radius) # server-private 2001:db8:a0b:12f0::1/64 timeout 5 RP/0//CPU0:router (config-sg-radius) # server-private 10.2.2.2 retransmit 3</pre>	Configures the IP address of the private RADIUS server for the group. If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used. Both auth-port and acct-port keywords enter RADIUS server-group private configuration mode.
Step 4	vrf <i>vrf-name</i> Example: <pre>RP/0//CPU0:router (config-sg-radius) # vrf v2.44.com</pre>	Configures the VRF reference of an AAA RADIUS server group. Note Private server IP addresses can overlap with those configured globally and the VRF definitions can help to distinguish them.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring a TACACS+ Server

This task configures a TACACS+ server.

The port, if not specified, defaults to the standard port number, 49. The **timeout** and **key** parameters can be specified globally for all TACACS+ servers. The **timeout** parameter specifies how long the AAA server waits to receive a response from the TACACS+ server. The **key** parameter specifies an authentication and encryption key shared between the AAA server and the TACACS+ server.

SUMMARY STEPS

1. **configure**
2. **tacacs-server host** *host-name* **port** *port-number*
3. **tacacs-server host** *host-name* **timeout** *seconds*
4. **tacacs-server host** *host-name* **key** [**0** | **7**] *auth-key*
5. **tacacs-server host** *host-name* **single-connection**
6. **tacacs source-interface** *type* *instance*
7. Repeat [Step 2, on page 36](#) through [Step 5, on page 37](#) for each external server to be configured.
8. Use the **commit** or **end** command.
9. **show tacacs**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	tacacs-server host <i>host-name</i> port <i>port-number</i> Example: RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 port 51 RP/0/RP0/CPU0:router(config-tacacs-host)#	Specifies a TACACS+ host server and optionally specifies a server port number. <ul style="list-style-type: none"> • This option overrides the default, port 49. Valid port numbers range from 1 to 65535.
Step 3	tacacs-server host <i>host-name</i> timeout <i>seconds</i> Example: RP/0/RP0/CPU0:router(config-tacacs-host)# tacacs-server host 209.165.200.226 timeout 30 RP/0/RP0/CPU0:router(config)#	Specifies a TACACS+ host server and optionally specifies a timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ server. <ul style="list-style-type: none"> • This option overrides the global timeout value set with the tacacs-server timeout command for only this server. The timeout value is expressed as an integer in terms of timeout interval seconds. The range is from 1 to 1000.
Step 4	tacacs-server host <i>host-name</i> key [0 7] <i>auth-key</i> Example: RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 key 0 a_secret	Specifies a TACACS+ host server and optionally specifies an authentication and encryption key shared between the AAA server and the TACACS+ server. <ul style="list-style-type: none"> • The TACACS+ packets are encrypted using this key. This key must match the key used by TACACS+ daemon. Specifying this key overrides the global key

	Command or Action	Purpose
		<p>set by the tacacs-server key command for only this server.</p> <ul style="list-style-type: none"> • (Optional) Entering 0 indicates that an unencrypted (clear-text) key follows. • (Optional) Entering 7 indicates that an encrypted key follows. • The <i>auth-key</i> argument specifies the encrypted or unencrypted key to be shared between the AAA server and the TACACS+ server.
Step 5	<p>tacacs-server host <i>host-name</i> single-connection</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 single-connection</pre>	<p>Prompts the router to multiplex all TACACS+ requests to this server over a single TCP connection. By default, a separate connection is used for each session.</p>
Step 6	<p>tacacs source-interface <i>type instance</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# tacacs source-interface 0/4/0/0</pre>	<p>(Optional) Specifies the source IP address of a selected interface for all outgoing TACACS+ packets.</p> <ul style="list-style-type: none"> • The specified interface or subinterface must have an IP address associated with it. If the specified interface or subinterface does not have an IP address or is in the down state, then TACACS+ reverts to the default interface. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state. • The vrf option specifies the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an AAA TACACS+ server group.
Step 7	Repeat Step 2, on page 36 through Step 5, on page 37 for each external server to be configured.	—
Step 8	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 9	<p>show tacacs</p> <p>Example:</p>	(Optional) Displays information about the TACACS+ servers that are configured in the system.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# show tacacs	

What to do next

After configuring TACACS+ servers, configure TACACS+ server groups. (See the [Configuring TACACS+ Server Groups, on page 40](#) section.)

Configuring RADIUS Server Groups

This task configures RADIUS server groups.

The user can enter one or more **server** commands. The **server** command specifies the hostname or IP address of an external RADIUS server along with port numbers. When configured, this server group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting). (See the [Method Lists, on page 10](#) section.)

Before you begin

For configuration to succeed, the external server should be accessible at the time of configuration.

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *group-name*
3. **server** {*hostname* | *ip address in IPv4 or IPv6 format*} [**auth-port** *port-number*] [**acct-port** *port-number*]
4. Repeat [Step 4, on page 39](#) for every external server to be added to the server group named in [Step 3, on page 38](#).
5. **deadtime** *minutes*
6. Use the **commit** or **end** command.
7. **show radius server-groups** [*group-name* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	aaa group server radius <i>group-name</i> Example: RP/0/RP0/CPU0:router(config)# aaa group server radius radgroup1	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	server { <i>hostname</i> <i>ip address in IPv4 or IPv6 format</i> } [auth-port <i>port-number</i>] [acct-port <i>port-number</i>] Example:	Specifies the hostname or IP address of an external RADIUS server. <ul style="list-style-type: none"> • After the server group is configured, it can be referenced from the AAA method lists (used while

	Command or Action	Purpose
	<p>IP address in IPv4 format</p> <pre>RP/0/RP0/CPU0:router(config-sg-radius)# server 192.168.20.0</pre> <p>Example:</p> <p>IP address in IPv6 format</p> <pre>RP/0/RP0/CPU0:router(config-sg-radius)# server 2001:db8:a0b:12f0::1/64</pre>	configuring authentication, authorization, or accounting).
Step 4	Repeat Step 4, on page 39 for every external server to be added to the server group named in Step 3, on page 38 .	—
Step 5	<p>deadtime <i>minutes</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-sg-radius)# deadtime 1</pre>	<p>Configures the deadtime value at the RADIUS server group level.</p> <ul style="list-style-type: none"> The <i>minutes</i> argument specifies the length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 (24 hours). The range is from 1 to 1440. <p>The example specifies a one-minute deadtime for RADIUS server group radgroup1 when it has failed to respond to authentication requests for the deadtime command</p> <p>Note You can configure the group-level deadtime after the group is created.</p>
Step 6	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session. No —Exits the configuration session without committing the configuration changes. Cancel —Remains in the configuration session, without committing the configuration changes.
Step 7	<p>show radius server-groups [<i>group-name</i> [detail]]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show radius server-groups</pre>	(Optional) Displays information about each RADIUS server group that is configured in the system.

What to do next

After configuring RADIUS server groups, define method lists by configuring authentication, authorization, and accounting. (See the [Configuring AAA Method Lists, on page 43](#) section.)

Configuring TACACS+ Server Groups

This task configures TACACS+ server groups.

You can enter one or more **server** commands. The **server** command specifies the hostname or IP address of an external TACACS+ server. Once configured, this server group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting). (See the [Method Lists, on page 10](#) section.)

Before you begin

For successful configuration, the external server should be accessible at the time of configuration. When configuring the same IP address for global and vrf configuration, server-private parameters are required.

SUMMARY STEPS

1. **configure**
2. **aaa group server tacacs+ *group-name***
3. **server {*hostname* | *ip address in IPv4 or IPv6 format*}**
4. Repeat [Step 3, on page 40](#) for every external server to be added to the server group named in [Step 2, on page 40](#).
5. Use the **commit** or **end** command.
6. **show tacacs server-groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	aaa group server tacacs+ <i>group-name</i> Example: RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	server {<i>hostname</i> <i>ip address in IPv4 or IPv6 format</i>} Example: IP address in IPv4 format RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 192.168.100.0 Example: IP address in IPv6 format RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 2001:db8:a0b:12f0::1/64	Specifies the hostname or IP address of an external TACACS+ server. <ul style="list-style-type: none">• When configured, this group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).
Step 4	Repeat Step 3, on page 40 for every external server to be added to the server group named in Step 2, on page 40 .	—

	Command or Action	Purpose
Step 5	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 6	<p>show tacacs server-groups</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show tacacs server-groups</pre>	(Optional) Displays information about each TACACS+ server group that is configured in the system.

What to do next

After configuring TACACS+ server groups, define method lists used by configuring authentication, authorization, and accounting. (See the [Configuring AAA Method Lists, on page 43](#) section.)

Configure Per VRF TACACS+ Server Groups

The Cisco IOS XR software supports per VRF AAA to be configured on TACACS+ server groups. You must use the **server-private** and **vrf** commands as listed below to configure this feature.

The global server definitions can be referred from multiple server groups, but all references use the same server instance and connect to the same server. In case of VRF, you do not need the global configuration because the server status, server statistics and the key could be different for different VRFs. Therefore, you must use the server-private configuration if you want to configure per VRF TACACS+ server groups. If you have the same server used in different groups with different VRFs, ensure that it is reachable through all those VRFs.

If you are migrating the servers to a VRF, then it is safe to remove the global server configuration with respect to that server.

Prerequisites

You must ensure these before configuring per VRF on TACACS+ server groups:

- Be familiar with configuring TACACS+, AAA, per VRF AAA, and group servers.
- Ensure that you have access to the TACACS+ server.
- Configure the VRF instance before configuring the specific VRF for a TACACS+ server and ensure that the VRF is reachable.

Configuration Example

```

Router#configure

/* Groups different server hosts into distinct lists and enters the server group configuration
mode.
You can enter one or more server commands. The server command specifies the hostname or IP
address of an external TACACS+ server.
Once configured, this server group can be referenced from the AAA method lists (used while
configuring authentication, authorization, or accounting). */

Router(config)# aaa group server tacacs+ tacgroup1

/* Configures the IP address and the secret key of the private TACACS+ server that is
reachable through specific VRF.
You can have multiple such server configurations which are reachable through the same VRF.*/

Router(config-sg-tacacs+)# server-private 10.1.1.1 port 49 key a_secret

/* The vrf option specifies the VRF reference of a AAA TACACS+ server group */
Router(config-sg-tacacs+)# vrf test-vrf
Router(config-sg-tacacs+)# commit

```

Running Configuration

```

aaa group server tacacs+ tacgroup1
vrf test-vrf
server-private 10.1.1.1 port 49
key 7 0822455D0A16
!
server-private 10.1.1.2 port 49
key 7 05080F1C2243
!
server-private 2001:db8:1::1 port 49
key 7 045802150C2E
!
server-private 2001:db8:1::2 port 49
key 7 13061E010803
!
!

```

Verify Per VRF TACACS+ Server Groups

```

Router#show tacacs
Fri Sep 27 11:14:34.991 UTC

Server: 10.1.1.1/49 vrf=test-vrf [private]
opens=0 closes=0 aborts=0 errors=0
packets in=0 packets out=0
status=up single-connect=false family=IPv4

Server: 10.1.1.2/49 vrf=test-vrf [private]
opens=0 closes=0 aborts=0 errors=0
packets in=0 packets out=0
status=up single-connect=false family=IPv4

Server: 2001:db8:1::1/49 vrf=test-vrf [private]
opens=0 closes=0 aborts=0 errors=0
packets in=0 packets out=0
status=up single-connect=false family=IPv6

```

```
Server: 2001:db8:1::2/49 vrf=test-vrf [private]
opens=0 closes=0 aborts=0 errors=0
packets in=0 packets out=0
status=up single-connect=false family=IPv6
```

Associated Commands

- **server-private**
- **vrf**

Configuring AAA Method Lists

AAA data may be stored in a variety of data sources. AAA configuration uses *method lists* to define an order of preference for the source of AAA data. AAA may define more than one method list and applications (such as login) can choose one of them. For example, console ports may use one method list and the vty ports may use another. If a method list is not specified, the application tries to use a default method list.

This section contains the following procedures:

Configuring Authentication Method Lists

This task configures method lists for authentication.

Authentication Configuration

Authentication is the process by which a user (or a principal) is verified. Authentication configuration uses *method lists* to define an order of preference for the source of AAA data, which may be stored in a variety of data sources. You can configure authentication to define more than one method list and applications (such as login) can choose one of them. For example, console ports may use one method list and the vty ports may use another. If a method list is not specified, the application tries to use a default method list.



Note Applications should explicitly refer to defined method lists for the method lists to be effective.

The authentication can be applied to tty lines through use of the **login authentication** line configuration submode command.

Creation of a Series of Authentication Methods

Use the **aaa authentication** command to create a series of authentication methods, or method list. A method list is a named list describing the authentication methods to be used (such as RADIUS or TACACS+), in sequence. The method will be one of the following:

- **group radius**—Use a server group or RADIUS servers for authentication
- **group tacacs+**—Use a server group or TACACS+ servers for authentication
- **local**—Use the local username or password database for authentication
- **line**—Use the line password or user group for authentication

If the method is RADIUS or TACACS+ servers, rather than server group, the RADIUS or TACACS+ server is chosen from the global pool of configured RADIUS and TACACS+ servers, in the order of configuration. Servers from this global pool are the servers that can be selectively added to a server group.

The subsequent methods of authentication are used only if the initial method returns an error, not if the request is rejected.

Before you begin



Note The default method list is applied for all the interfaces for authentication, except when a non-default named method list is explicitly configured, in which case the named method list is applied.

The **group radius**, **group tacacs+**, and **group group-name** forms of the **aaa authentication** command refer to a set of previously defined RADIUS or TACACS+ servers. Use the **radius server-host** or **tacacs-server host** command to configure the host servers. Use the **aaa group server radius** or **aaa group server tacacs+** command to create a named group of servers.

SUMMARY STEPS

1. **configure**
2. **aaa authentication {login} {default | list-name} method-list**
3. Use the **commit** or **end** command.
4. Repeat Step 1 through Step 3 for every authentication method list to be configured.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	aaa authentication {login} {default list-name} method-list Example: RP/0//CPU0:router(config)# aaa authentication login default group tacacs+	Creates a series of authentication methods, or a method list. <ul style="list-style-type: none"> • Using the login keyword sets authentication for login. Using the ppp keyword sets authentication for Point-to-Point Protocol. • Entering the default keyword causes the listed authentication methods that follow this keyword to be the default list of methods for authentication. • Entering a <i>list-name</i> character string identifies the authentication method list. • Entering a <i>method-list</i> argument following the method list type. Method list types are entered in the preferred sequence. The listed method types are any one of the following options:

	Command or Action	Purpose
		<ul style="list-style-type: none"> • group tacacs+—Use a server group or TACACS+ servers for authentication • group radius—Use a server group or RADIUS servers for authentication • group <i>named-group</i>—Use a named subset of TACACS+ or RADIUS servers for authentication • local—Use a local username or password database for authentication • line—Use line password or user group for authentication <p>• The example specifies the default method list to be used for authentication.</p>
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 4	Repeat Step 1 through Step 3 for every authentication method list to be configured.	—

What to do next

After configuring authentication method lists, configure authorization method lists. (See the [Configuring Authorization Method Lists, on page 45](#) section).

Configuring Authorization Method Lists

This task configures method lists for authorization.



Note You can configure the **radius** keyword for the **aaa authorization** command.

Authorization Configuration

Method lists for authorization define the ways authorization will be performed and the sequence in which these methods will be performed. A method list is a named list describing the authorization methods to be

used (such as TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system if the initial method fails. The software uses the first method listed to authorize users for specific network services; if that method fails to respond, the software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or until all methods defined have been exhausted.



Note The software attempts authorization with the next listed method only when there is no response or an error response (not a failure) from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the type of authorization being requested. Four types of AAA authorization are supported:

- **Commands authorization**—Applies to the XR EXEC mode mode commands a user issues. Command authorization attempts authorization for all XR EXEC mode mode commands.



Note “Command” authorization is distinct from “task-based” authorization, which is based on the task profile established during authentication.

- **XR EXEC mode authorization**—Applies authorization for starting XR EXEC mode session.
- **Network authorization**—Applies authorization for network services, such as IKE.
- **Eventmanager authorization**—Applies an authorization method for authorizing an event manager (fault manager). RADIUS servers are not allowed to be configured for the event manager (fault manager) authorization. You are allowed to use TACACS+ or locald.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type. When defined, method lists must be applied to specific lines or interfaces before any of the defined methods are performed. Do not use the names of methods, such as TACACS+, when creating a new method list.

“Command” authorization, as a result of adding a command authorization method list to a line template, is separate from, and is in addition to, “task-based” authorization, which is performed automatically on the router. The default behavior for command authorization is none. Even if a default method list is configured, that method list has to be added to a line template for it to be used.

The **aaa authorization commands** command causes a request packet containing a series of attribute value (AV) pairs to be sent to the TACACS+ daemon as part of the authorization process. The daemon can do one of the following:

- Accept the request as is.
- Refuse authorization.

Creation of a Series of Authorization Methods

Use the **aaa authorization** command to set parameters for authorization and to create named method lists defining specific authorization methods that can be used for each line or interface.

The Cisco IOS XR software supports the following methods for authorization:

- **none**—The router does not request authorization information; authorization is not performed over this line or interface.
- **local**—Uses local database for authorization.
- **group tacacs+**—Uses the list of all configured TACACS+ servers for authorization.
- **group radius**—Uses the list of all configured RADIUS servers for authorization.
- **group *group-name***—Uses a named subset of TACACS+ servers for authorization.

SUMMARY STEPS

1. **configure**
2. **aaa authorization {commands | eventmanager | exec | network} {default | list-name} {none | local | group {tacacs+ | radius | group-name}}**
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	<p>aaa authorization {commands eventmanager exec network} {default list-name} {none local group {tacacs+ radius group-name}}</p> <p>Example:</p> <pre>RP/0//CPU0:router(config)# aaa authorization commands listname1 group tacacs+</pre>	<p>Creates a series of authorization methods, or a method list.</p> <ul style="list-style-type: none"> • The commands keyword configures authorization for all XR EXEC mode shell commands. Command authorization applies to the EXEC mode commands issued by a user. Command authorization attempts authorization for all XR EXEC mode commands. • The eventmanager keyword applies an authorization method for authorizing an event manager (fault manager). • The exec keyword configures authorization for an interactive (XR EXEC mode) session. • The network keyword configures authorization for network services like PPP or IKE. • The default keyword causes the listed authorization methods that follow this keyword to be the default list of methods for authorization. • A <i>list-name</i> character string identifies the authorization method list. The method list itself follows the method list name. Method list types are entered in the preferred

	Command or Action	Purpose
		<p>sequence. The listed method list types can be any one of the following:</p> <ul style="list-style-type: none"> • none—The network access server (NAS) does not request authorization information. Authorization always succeeds. No subsequent authorization methods will be attempted. However, the task ID authorization is always required and cannot be disabled. • local—Uses local database for authorization. • group tacacs+—Uses the list of all configured TACACS+ servers for authorization. The NAS exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating AV pairs, which are stored in a database on the TACACS+ security server, with the appropriate user. • group radius—Uses the list of all configured RADIUS servers for authorization. • group <i>group-name</i>—Uses a named server group, a subset of TACACS+ or RADIUS servers for authorization as defined by the aaa group server tacacs+ or aaa group server radius command.
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After configuring authorization method lists, configure accounting method lists. (See the [Configuring Accounting Method Lists, on page 48](#) section.)

Configuring Accounting Method Lists

This task configures method lists for accounting.



Note You can configure the **radius** keyword for the **aaa accounting** command.

Accounting Configuration

Currently, Cisco IOS XR software supports both the TACACS+ and RADIUS methods for accounting. The router reports user activity to the TACACS+ or RADIUS security server in the form of accounting records. Each accounting record contains accounting AV pairs and is stored on the security server.

Method lists for accounting define the way accounting is performed, enabling you to designate a particular security protocol to be used on specific lines or interfaces for particular types of accounting services. When naming a method list, do not use the names of methods, such as TACACS+.

For minimal accounting, include the **stop-only** keyword to send a “stop accounting” notice at the end of the requested user process. For more accounting, you can include the **start-stop** keyword, so that the external AAA server sends a “start accounting” notice at the beginning of the requested process and a “stop accounting” notice at the end of the process. In addition, you can use the **aaa accounting update** command to periodically send update records with accumulated information. Accounting records are stored only on the TACACS+ or RADIUS server.

When AAA accounting is activated, the router reports these attributes as accounting records, which are then stored in an accounting log on the security server.

Creation of a Series of Accounting Methods

Use the **aaa accounting** command to create default or named method lists defining specific accounting methods that can be used for each line or interface.

The Cisco IOS XR software supports the following methods for accounting:

- none—Accounting is not performed over this line or interface.
- group tacacs+—Use the list of all configured TACACS+ servers for accounting.
- group radius—Use the list of all configured RADIUS servers for accounting.

SUMMARY STEPS

1. **configure**
2. Do one of the following:
 - **aaa accounting** {**commands** | **exec** | **network**} {**default** | *list-name*} {**start-stop** | **stop-only**}
 - {**none** | *method*}
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters XR Config mode.

	Command or Action	Purpose
<p>Step 2</p>	<p>Do one of the following:</p> <ul style="list-style-type: none"> • aaa accounting {commands exec network} {default <i>list-name</i>} {start-stop stop-only} • {none <i>method</i>} <p>Example:</p> <pre>RP/0//CPU0:router(config)# aaa accounting commands default stop-only group tacacs+</pre>	<p>Note Command accounting is not supported on RADIUS, but supported on TACACS.</p> <p>Creates a series of accounting methods, or a method list.</p> <ul style="list-style-type: none"> • The commands keyword enables accounting for XR EXEC mode shell commands. • The exec keyword enables accounting for an interactive (XR EXEC mode) session. • The network keyword enables accounting for all network-related service requests, such as Point-to-Point Protocol (PPP). • The default keyword causes the listed accounting methods that follow this keyword to be the default list of methods for accounting. • A <i>list-name</i> character string identifies the accounting method list. • The start-stop keyword sends a “start accounting” notice at the beginning of a process and a “stop accounting” notice at the end of a process. The requested user process begins regardless of whether the “start accounting” notice was received by the accounting server. • The stop-only keyword sends a “stop accounting” notice at the end of the requested user process. • The none keyword states that no accounting is performed. • The method list itself follows the start-stop keyword. Method list types are entered in the preferred sequence. The method argument lists the following types: <ul style="list-style-type: none"> • group tacacs+—Use the list of all configured TACACS+ servers for accounting. • group radius—Use the list of all configured RADIUS servers for accounting. • group group-name—Use a named server group, a subset of TACACS+ or RADIUS servers for accounting as defined by the aaa group server tacacs+ or aaa group server radius command. • The example defines a default command accounting method list, in which accounting services are provided by a TACACS+ security server, with a stop-only restriction.

	Command or Action	Purpose
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After configuring method lists, apply those method lists. (See the [Applying Method Lists for Applications](#), on page 52 section.)

Generating Interim Accounting Records

This task enables periodic interim accounting records to be sent to the accounting server. When the **aaa accounting update** command is activated, Cisco IOS XR software issues interim accounting records for all users on the system.



Note Interim accounting records are generated only for network sessions, such as Internet Key Exchange (IKE) accounting, which is controlled by the **aaa accounting** command with the **network** keyword. System, command, or EXEC accounting sessions cannot have interim records generated.

SUMMARY STEPS

1. **configure**
2. **aaa accounting update {newinfo | periodic minutes}**
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	<p>aaa accounting update {newinfo periodic minutes}</p> <p>Example:</p>	<p>Enables periodic interim accounting records to be sent to the accounting server.</p> <ul style="list-style-type: none"> • If the newinfo keyword is used, interim accounting records are sent to the accounting server every time

	Command or Action	Purpose
	RP/0//CPU0:router(config)# aaa accounting update periodic 30	<p>there is new accounting information to report. An example of this report would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record includes the negotiated IP address used by the remote peer.</p> <ul style="list-style-type: none"> • When used with the periodic keyword, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all the accounting information recorded for that user up to the time the interim accounting record is sent. <p>Caution The periodic keyword causes heavy congestion when many users are logged in to the network.</p>
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Applying Method Lists for Applications

After you configure method lists for authorization and accounting services, you can apply those method lists for applications that use those services (console, vty, and so on). Applying method lists is accomplished by enabling AAA authorization and accounting.

This section contains the following procedures:

Enabling AAA Authorization

This task enables AAA authorization for a specific line or group of lines.

Method List Application

After you use the **aaa authorization** command to define a named authorization method list (or use the default method list) for a particular type of authorization, you must apply the defined lists to the appropriate lines in order for authorization to take place. Use the **authorization** command to apply the specified method lists (or, if none is specified, the default method list) to the selected line or group of lines.

SUMMARY STEPS

1. **configure**
2. **line** { **console** | **default** | **template** *template-name*}
3. **authorization** {**commands** | **exec**} {**default** | *list-name*}
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	line { console default template <i>template-name</i> } Example: RP/0//CPU0:router(config)# line console	Enters line template configuration mode.
Step 3	authorization { commands exec } { default <i>list-name</i> } Example: RP/0//CPU0:router(config-line)# authorization commands listname5	Enables AAA authorization for a specific line or group of lines. <ul style="list-style-type: none"> • The commands keyword enables authorization on the selected lines for all commands. • The exec keyword enables authorization for an interactive (XR EXEC mode) session. • Enter the default keyword to apply the name of the default method list, as defined with the aaa authorization command. • Enter the name of a list of authorization methods to use. If no list name is specified, the system uses the default. The list is created with the aaa authorization command. • The example enables command authorization using the method list named listname5.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After applying authorization method lists by enabling AAA authorization, apply accounting method lists by enabling AAA accounting. (See the [Enabling Accounting Services, on page 54](#) section.)

Enabling Accounting Services

This task enables accounting services for a specific line of group of lines.

SUMMARY STEPS

1. **configure**
2. **line { console | default | template template-name}**
3. **accounting {commands | exec} {default | list-name}**
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	line { console default template template-name} Example: RP/0//CPU0:router(config)# line console	Enters line template configuration mode.
Step 3	accounting {commands exec} {default list-name} Example: RP/0//CPU0:router(config-line)# accounting commands listname7	Enables AAA accounting for a specific line or group of lines. <ul style="list-style-type: none"> • The commands keyword enables accounting on the selected lines for all XR EXEC mode shell commands. • The exec keyword enables accounting for an interactive (XR EXEC mode) session. • Enter the default keyword to apply the name of the default method list, as defined with the aaa accounting command. • Enter the name of a list of accounting methods to use. If no list name is specified, the system uses the default. The list is created with the aaa accounting command. • The example enables command accounting using the method list named listname7.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions:

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No — Exits the configuration session without committing the configuration changes. • Cancel — Remains in the configuration session, without committing the configuration changes.

What to do next

After applying accounting method lists by enabling AAA accounting services, configure login parameters. (See the [Configuring Login Parameters, on page 55](#) section.)

Configuring Login Parameters

This task sets the interval that the server waits for reply to a login.

SUMMARY STEPS

1. **configure**
2. **line template** *template-name*
3. **timeout login response** *seconds*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	line template <i>template-name</i> Example: RP/0//CPU0:router(config)# line template alpha	Specifies a line to configure and enters line template configuration mode.
Step 3	timeout login response <i>seconds</i> Example: RP/0//CPU0:router(config-line)# timeout login response 20	Sets the interval that the server waits for reply to a login. <ul style="list-style-type: none"> • The <i>seconds</i> argument specifies the timeout interval (in seconds) from 0 to 300. The default is 30 seconds. • The example shows how to change the interval timer to 20 seconds.
Step 4	Use the commit or end command.	commit — Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		<p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring a TACACS+ Server on the Sysadmin Plane

This task describes how to configure TACACS+server on the Sysadmin plane.

Configuring TACACS+ Server

This task describes how to configure TACACS+server on the Sysadmin plane, specify an authorization, encryption key, and a timeout value to indicate the length of time the AAA server waits to receive a response from the TACACS+ server to establish remote-user login with AAA subsystems

SUMMARY STEPS

1. **configure**
2. **tacacs-server host** *host ip address port number*
3. **tacacs-server host** *host ip address port number timeout value in seconds*
4. **tacacs-server timeout** *value in seconds*
5. **tacacs-server host** *host ip address port number key auth-key*
6. **tacacs-server key** *value*
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	<p>tacacs-server host <i>host ip address port number</i></p> <p>Example:</p> <pre>sysadmin-vm:0_RP0(config)# tacacs-server host 209.165.200.226 51</pre>	Specifies a TACACS+ host server and optionally specify a server port number. This option overrides the default, port 49. Valid port numbers range from 1 to 65535.

	Command or Action	Purpose
Step 3	<p>tacacs-server host <i>host ip address port number timeout value in seconds</i></p> <p>Example:</p> <pre>sysadmin-vm:0_RP0(config)# tacacs-server host 209.165.200.226 51 timeout 60</pre>	Specifies a TACACS+ host server and optionally specifies a timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ server. This option overrides the global timeout value set with the tacacs-server timeout command for only this server. The timeout value is expressed as an integer in terms of timeout interval seconds. The range is from 1 to 1000.
Step 4	<p>tacacs-server timeout <i>value in seconds</i></p> <p>Example:</p> <pre>sysadmin-vm:0_RP0(config)# tacacs-server 209.165.200.227 timeout 60</pre>	Specifies a TACACS+ global server and optionally specifies a timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ global server.
Step 5	<p>tacacs-server host <i>host ip address port number key auth-key</i></p> <p>Example:</p> <pre>sysadmin-vm:0_RP0(config)# tacacs-server host 209.165.200.226 51 key cisco_auth-key</pre>	Specifies a TACACS+ host server and optionally specifies an authentication and encryption key shared between the AAA server and the TACACS+ server. The TACACS+ packets are encrypted using this key. This key must match the key used by TACACS+ daemon. Specifying this key overrides the global key set by the tacacs-server key command for only this server.
Step 6	<p>tacacs-server key <i>value</i></p> <p>Example:</p> <pre>sysadmin-vm:0_RP0(config)# tacacs-server 209.165.200.227 cisco_server_key</pre>	Specifies a TACACS+ global server and specifies an authentication and encryption key shared between the AAA server and the TACACS+ global server. The TACACS+ packets are encrypted using this key.
Step 7	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring Authorization for a TACACS+ Server

This task helps you configure authorization commands are used to verify that an authenticated user (or principal) is granted permission to perform a specific task.

SUMMARY STEPS

1. **configure**
2. **aaa authorization command group tacacs | none**

3. **confdConfig aaa authorization enabled**
4. **confdConfig aaa authorization callback enabled**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	aaa authorization command group tacacs none Example: sysadmin-vm:0_RP0(config)# aaa authorization command group tacacs Example: sysadmin-vm:0_RP0(config)# aaa authorization command group none Example: sysadmin-vm:0_RP0(config)# aaa authorization command group tacacs none	Configure the AAA system to perform remote authorization using TACACS+ protocol. Configure the AAA system to not perform any authorization. Configure the AAA system to first perform TACACS+ authorization and if it fails, no authorization should be performed.
Step 3	confdConfig aaa authorization enabled Example: sysadmin-vm:0_RP0(config)# confdConfig aaa authorization enabled	Configure ConfD to perform remote authorization.
Step 4	confdConfig aaa authorization callback enabled Example: sysadmin-vm:0_RP0(config)# confdConfig aaa authorization callback enabled	Configure ConfD to invoke application callbacks for authorization.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring Authentication for a TACACS+ Server

This task describes how to configure authentication commands to verify the identity of a user or principal TACACS+server on Sysadmin plane.

SUMMARY STEPS

1. **configure**
2. **confdConfig aaa externalAuthentication enabled**
3. **confdConfig aaa authOrder localAuthentication|externalAuthentication**
4. **confdConfig aaa externalAuthentication executable"chvrf 0 /opt/cisco/calvados/bin/calvados_login_aaa_proxy"**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	confdConfig aaa externalAuthentication enabled Example: sysadmin-vm:0_RP0(config)# confdConfig aaa externalAuthentication enabled	Configure Confd to perform external authentication.
Step 3	confdConfig aaa authOrder localAuthentication externalAuthentication Example: sysadmin-vm:0_RP0(config)# confdConfig aaa authOrder externalAuthentication localAuthentication	Configure the AAA subsystem to perform external authentication first and then local authentication.
Step 4	confdConfig aaa externalAuthentication executable"chvrf 0 /opt/cisco/calvados/bin/calvados_login_aaa_proxy" Example: sysadmin-vm:0_RP0(config)# confdConfig aaa externalAuthentication executable chvrf 0 /opt/cisco/calvados/bin/calvados_login_aaa_proxy	Configure the AAA system to perform external authentication using login executable configured on local host.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring Accounting for a TACACS+ Server

This task describes how to configure accounting commands that are used for logging of sessions and to create an audit trail by recording certain user- or system-generated actions.

SUMMARY STEPS

1. **configure**
2. **aaa accounting command tacacs**
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	aaa accounting command tacacs Example: sysadmin-vm:0_RP0(config)# aaa accounting command tacacs	Configure remote accounting commands.
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuration Examples for Configuring AAA Services

This section provides the following configuration example:

Configuring AAA Services: Example

The following examples show how to configure AAA services.

An authentication method list vty-authen is configured. This example specifies a method list that uses the list of all configured TACACS+ servers for authentication. If that method fails, the local username database method is used for authentication.

```
configure
aaa authentication login vty-authen group tacacs+ local
```

The default method list for PPP is configured to use local method.

```
aaa authentication ppp default local
```

A username user1 is created for login purposes, a secure login password is assigned, and user1 is made a root-lr user. Configure similar settings for username user2.

```
username user1
secret lab
group root-lr
exit
```

```
username user2
secret lab
exit
```

A task group named tga is created, tasks are added to tga, a user group named uga is created, and uga is configured to inherit permissions from task group tga. A description is added to task group uga.

```
taskgroup tga
task read bgp
task write ospf
exit
```

```
usergroup uga
taskgroup tga
description usergroup uga
exit
```

Username user2 is configured to inherit from user group uga.

```
username user2
group uga
exit
```

Three TACACS servers are configured.

```
tacacs-server host .1.1.1 port 1 key abc
tacacs-server host .2.2.2 port 2 key def
tacacs-server host .3.3.3 port 3 key ghi
```

A user group named priv5 is created, which will be used for users authenticated using the TACACS+ method and whose entry in the external TACACS+ daemon configuration file has a privilege level of 5.

```
usergroup priv5
taskgroup operator
exit
```

An authorization method list, `vty-author`, is configured. This example specifies that command authorization be done using the list of all configured TACACS+ servers.

```
aaa authorization commands vty-author group tacacs+
```

An accounting method list, `vty-acct`, is configured. This example specifies that start-stop command accounting be done using the list of all configured TACACS+ servers.

```
aaa accounting commands vty-acct start-stop group tacacs+
```

For TACACS+ authentication, if, for example, a privilege level 8 is returned, and no local usergroup `priv8` exists and no local user with the same name exists, the **aaa default-taskgroup** command with `tga` specified as the *taskgroup-name* argument ensures that such users are given the taskmap of the task group `tga`.

```
aaa default-taskgroup tga
```

For line template `vty`, a line password is assigned that is used with line authentication and makes usergroup `uga` the group that is assigned for line authentication (if used), and makes `vty-authen`, `vty-author`, and `vty-acct`, respectively, the method lists that are used for authentication, authorization, and accounting.

```
line template vty
password lab
users group uga
login authentication vty-authen
authorization commands vty-author
accounting commands vty-acct
exit
```

A TACACS+ server group named `abc` is created and an already configured TACACS+ server is added to it.

```
aaa group server tacacs+ abc
server .3.3.3
exit
```

Feature Overview

The NETCONF protocol does not provide any standard mechanisms to restrict the protocol operations and content that each user is authorized to access. NACM is defined in AAA subsystem to manage access-control for NETCONF/YANG RPC requests.

Prerequisites for Model Based AAA

Working with the model based AAA feature requires prior understanding of the following :

- NETCONF-YANG
- RFC 6536: Network Configuration Protocol (NETCONF) Access Control Model

Initial Operation

These are the NACM default values. By default a user is denied write permission, hence you'll not be able to edit the NACM configurations after enabling NACM authorization using AAA command.

```

<enable-nacm>true</enable-nacm>
<read-default>permit</read-default>
<write-default>deny</write-default>
<exec-default>permit</exec-default>
<enable-external-groups>true</enable-external-groups>

```

Therefore we recommend to enable NACM after configuring the required NACM configurations, or after changing the default NACM configurations. Here are few sample configurations:



Note If an user gets access-denied message while writing NACM configurations, then NACM authorization can be disabled to edit the NACM configurations.

```

<aaa xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-lib-cfg">
<usernames xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-locald-cfg">
<username>
<ordering-index>3</ordering-index>
<name>username</name>
<password>password</password>
  <usergroup-under-usernames>
    <usergroup-under-username>
      <name>root-lr</name>
    </usergroup-under-username>
    <usergroup-under-username>
      <name>cisco-support</name>
    </usergroup-under-username>
  </usergroup-under-usernames>
</username>
</usernames>
</aaa>

<nacm xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-nacm-cfg">
<read-default>permit</read-default>
<write-default>permit</write-default>
<exec-default>permit</exec-default>
<enable-external-groups>true</enable-external-groups>
<groups>
  <group>
    <name>nacm_group</name>
    <user-name>lab</user-name>
  </group>
</groups>
<rule-list>
<name>Rule-list-1</name>
<group>Group_nacm_0_test</group>
<rule>
  <name>Rule-1</name>
  <access-operations>read</access-operations>
  <action>permit</action>
  <module-name>ietf-netconf-acm</module-name>
  <rpc-name>edit-config</rpc-name>
    <access-operations>*</access-operations>
    <path>/</path>
  <action>permit</action>
</rule>
</rule-list>
</nacm>

```

NACM Configuration Management and Persistence

The NACM configuration can be modified using NETCONF or RESTCONF. In order for a user to be able to access the NACM configuration, they must have explicit permission to do so, that is, through a NACM rule. Configuration under the /nacm subtree persists when the **copy running-config startup-config EXEC** command is issued, or the **cisco-ia:save-config** RPC is issued.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<save-config xmlns="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

Overview of Configuring NACM

Here are the steps involved in configuring NACM:

1. Configure all NACM rules
2. Enable NACM
3. Disconnect all active NETCONF sessions
4. Launch new NETCONF session



Note Enabling or disabling NACM doesn't affect any existing NETCONF sessions.

NACM Rules

As per the RFC 6536, NACM defines two categories of rules:

- Global Rules—It includes the following:
 - Enable/Disable NACM
 - Read-Default
 - Write-Default
 - Exec-Default
 - Enable external Groups
- Access Control Rules—It includes the following:
 - Module (used along with protocol rule / data node rule)
 - Protocol
 - Data Node

Configuring Global Rules

You must configure NACM groups and NACM rulelist before configuring NACM rules.

The following sample configuration shows a NACM group configuration:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" >
<edit-config>
  <target><candidate/></target>
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <groups>
      <group>
        <name>group1</name>
        <user-name>user1</user-name>
        <user-name>user2</user-name>
        <user-name>user3</user-name>
      </group>
    </groups>
  </nacm>
</config>
</edit-config>
</rpc>
```

The following sample configuration shows a NACM rulelist configuration:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" >
<edit-config>
<target><candidate/></target>
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <rule-list>
      <name>rulelist-1</name>
      <group>root-lr</group>
      <group>group1</group>
      <group>group2</group>
    </rule-list>
  </nacm>
</config>
</edit-config>
</rpc>
```

This is the sample for configuring the NACM global rules:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" >
<edit-config>
  <target><candidate/></target>
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <read-default>permit</read-default>
    <write-default>permit</write-default>
    <exec-default>permit</exec-default>
    <enable-external-groups>false</enable-external-groups>
  </nacm>
</config>
</edit-config>
</rpc>
```

Configuring Access Control Rules

This is the sample configuration for the NACM protocol rules:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" >
<edit-config>
<target><candidate/></target>
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
```

```

<rule-list>
  <name>rulelist-1</name>
  <rule>
    <name>rule1</name>
    <module-name>ietf-netconf-acm</module-name>
    <rpc-name>edit-config</rpc-name>
    <access-operations>*</access-operations>
    <action>permit</action>
  </rule>
  <rule>
    <name>rule2</name>
    <module-name>ietf-netconf-acm</module-name>
    <rpc-name>get-config</rpc-name>
    <access-operations>create read update exec</access-operations>
    <action>permit</action>
  </rule>
</rule-list>
</nacm>
</config>
</edit-config>
</rpc>

```



Note '*' refers to all operations.

This is the sample configuration for the NACM data node rules:

```

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" >
<edit-config>
<target><candidate/></target>
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <rule-list>
      <name>rulelist-2</name>
      <rule>
        <name>rule4</name>
        <module-name>*</module-name>
        <path>/nacm/groups/group</path>
        <access-operations>*</access-operations>
        <action>permit</action>
      </rule>
      <rule>
        <name>rule5</name>
        <module-name>ietf-netconf-acm</module-name>
        <path>/nacm/rule-list</path>
        <access-operations>read</access-operations>
        <action>deny</action>
      </rule>
    </rule-list>
  </nacm>
</config>
</edit-config>
</rpc>

```



Note '*' refers to all modules, and all operations.

Enabling NACM

To enable NACM, use the **aaa authorization nacm default local** command in configuration mode:

```
Router(config)#aaa authorization nacm default local
```

Verification

Use the **show nacm summary** command to verify the default values after enabling NACM:

```
Router# show nacm summary
Mon Jan 15 16:47:43.549 UTC
NACM SUMMARY
-----
Enable Nacm : True
Enable External Groups : True
Number of Groups : 0
Number of Users : 0
Number of Rules : 0
Number of Rulelist : 0
Default Read : permit
Default Write : deny
Default Exec : permit
Denied Operations : 0
Denied Data Writes : 0
Denied Notifications : 0
```

Verify the NACM Configurations

Use the **show nacm summary** command to verify the NACM configurations:

```
Router# show nacm summary
Mon Jan 15 17:02:46.696 UTC
NACM SUMMARY
-----
Enable Nacm : True
Enable External Groups : True
Number of Groups : 3
Number of Users : 3
Number of Rules : 4
Number of Rulelist : 2
Default Read : permit
Default Write : permit
Default Exec : permit
Denied Operations : 1
Denied Data Writes : 0
Denied Notifications : 0
-----
```

Disabling NACM

There are two ways you can disable NACM:

1. Configuring NACM authorization as none:

```
Router(config)# aaa authorization nacm default none
```

2. Using no form of AAA authorization command:

```
(config)# no aaa authorization nacm default
```

Verification

Use the **show nacm summary** command to verify the default values after disabling NACM:

```
Router# show nacm summary
```

```
Mon Jan 15 17:02:46.696 UTC
NACM SUMMARY
```

```
-----
Enable Nacm : False
```

```
Enable External Groups : True
Number of Groups : 0
Number of Users : 0
Number of Rules : 0
Number of Rulelist : 0
Default Read : permit
Default Write : deny
Default Exec : permit
Denied Operations : 0
Denied Data Writes : 0
Denied Notifications : 0
```

Additional References

The following sections provide references related to configuring AAA services.

Related Documents

Related Topic	Document Title
AAA services commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Authentication, Authorization, and Accounting Commands on</i> in the

Standards

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

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

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

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 3

Implementing Certification Authority Interoperability

CA interoperability permits Cisco NCS 6000 Series Router devices and CAs to communicate so that your device can obtain and use digital certificates from the CA. Although IPSec can be implemented in your network without the use of a CA, using a CA provides manageability and scalability for IPSec.



Note For a complete description of the public key infrastructure (PKI) commands used in this chapter, refer to the *Public Key Infrastructure Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Implementing Certification Authority Interoperability

Release	Modification
Release 5.0.0	This feature was introduced.
Release 5.2	A section was added on trust pool management

- [Prerequisites for Implementing Certification Authority, on page 71](#)
- [Restrictions for Implementing Certification Authority, on page 72](#)
- [Information About Implementing Certification Authority, on page 72](#)
- [How to Implement CA Interoperability, on page 75](#)
- [Configuration Examples for Implementing Certification Authority Interoperability, on page 82](#)
- [Where to Go Next, on page 84](#)
- [Additional References, on page 84](#)

Prerequisites for Implementing Certification Authority

The following prerequisites are required to implement CA interoperability:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- You must install and activate the Package Installation Envelope (PIE) for the security software.

For detailed information about optional PIE installation, refer to *System Management Configuration Guide for Cisco NCS 6000 Series Routers*

- You need to have a CA available to your network before you configure this interoperability feature. The CA must support Cisco Systems PKI protocol, the simple certificate enrollment protocol (SCEP) (formerly called certificate enrollment protocol [CEP]).

Restrictions for Implementing Certification Authority

Cisco IOS XR software does not support CA server public keys greater than 2048 bits.

Information About Implementing Certification Authority

To implement CA, you need to understand the following concepts:

Supported Standards for Certification Authority Interoperability

Cisco supports the following standards:

- **IPSec**—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, a pair of security gateways, or a security gateway and a host.
- **IKE**—A hybrid protocol that implements Oakley and Skeme key exchanges inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations (SAs).
- **Public-Key Cryptography Standard #7 (PKCS #7)**—A standard from RSA Data Security Inc. used to encrypt and sign certificate enrollment messages.
- **Public-Key Cryptography Standard #10 (PKCS #10)**—A standard syntax from RSA Data Security Inc. for certificate requests.
- **RSA keys**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adelman. RSA keys come in pairs: one public key and one private key.
- **SSL**—Secure Socket Layer protocol.
- **X.509v3 certificates**—Certificate support that allows the IPSec-protected network to scale by providing the equivalent of a digital ID card to each device. When two devices want to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or specify a shared key at each peer). These certificates are obtained from a CA. X.509 as part of the X.500 standard of the ITU.

Certification Authorities

The following sections provide background information about CAs:

Purpose of CAs

CAs are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized key management for the participating devices.

CAs simplify the administration of IPSec network devices. You can use a CA with a network containing multiple IPSec-compliant devices, such as routers.

Digital signatures, enabled by public key cryptography, provide a means of digitally authenticating devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user's private key. The receiver verifies the signature by decrypting the message with the sender's public key. The fact that the message could be decrypted using the sender's public key indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver's having a copy of the sender's public key and knowing with a high degree of certainty that it does belong to the sender and not to someone pretending to be the sender.

Digital certificates provide the link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The certificate is itself signed by a CA, a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Normally, this process is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. IKE, an essential component of IPSec, can use digital signatures to authenticate peer devices for scalability before setting up SAs.

Without digital signatures, a user must manually exchange either public keys or secrets between each pair of devices that use IPSec to protect communication between them. Without certificates, every new device added to the network requires a configuration change on every other device with which it communicates securely. With digital certificates, each device is enrolled with a CA. When two devices want to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, a user simply enrolls that device with a CA, and none of the other devices needs modification. When the new device attempts an IPSec connection, certificates are automatically exchanged and the device can be authenticated.

IPSec Without CAs

Without a CA, if you want to enable IPSec services (such as encryption) between two Cisco routers, you must first ensure that each router has the key of the other router (such as an RSA public key or a shared key). This requirement means that you must manually perform one of the following operations:

- At each router, enter the RSA public key of the other router.
- At each router, specify a shared key to be used by both routers.

If you have multiple Cisco routers in a mesh topology and want to exchange IPSec traffic passing among all of those routers, you must first configure shared keys or RSA public keys among all of those routers.

Every time a new router is added to the IPSec network, you must configure keys between the new router and each of the existing routers.

Consequently, the more devices there are that require IPSec services, the more involved the key administration becomes. This approach does not scale well for larger, more complex encrypting networks.

IPSec with CAs

With a CA, you need not configure keys between all the encrypting routers. Instead, you individually enroll each participating router with the CA, requesting a certificate for the router. When this enrollment has been accomplished, each participating router can dynamically authenticate all the other participating routers.

To add a new IPSec router to the network, you need only configure that new router to request a certificate from the CA, instead of making multiple key configurations with all the other existing IPSec routers.

IPSec with Multiple Trustpoint CAs

With multiple trustpoint CAs, you no longer have to enroll a router with the CA that issued a certificate to a peer. Instead, you configure a router with multiple CAs that it trusts. Thus, a router can use a configured CA (a trusted root) to verify certificates offered by a peer that were not issued by the same CA defined in the identity of the router.

Configuring multiple CAs allows two or more routers enrolled under different domains (different CAs) to verify the identity of each other when using IKE to set up IPSec tunnels.

Through SCEP, each router is configured with a CA (the enrollment CA). The CA issues a certificate to the router that is signed with the private key of the CA. To verify the certificates of peers in the same domain, the router is also configured with the root certificate of the enrollment CA.

To verify the certificate of a peer from a different domain, the root certificate of the enrollment CA in the domain of the peer must be configured securely in the router.

During IKE phase one signature verification, the initiator will send the responder a list of its CA certificates. The responder should send the certificate issued by one of the CAs in the list. If the certificate is verified, the router saves the public key contained in the certificate on its public key ring.

With multiple root CAs, Virtual Private Network (VPN) users can establish trust in one domain and easily and securely distribute it to other domains. Thus, the required private communication channel between entities authenticated under different domains can occur.

How IPSec Devices Use CA Certificates

When two IPSec routers want to exchange IPSec-protected traffic passing between them, they must first authenticate each other—otherwise, IPSec protection cannot occur. The authentication is done with IKE.

Without a CA, a router authenticates itself to the remote router using either RSA-encrypted nonces or preshared keys. Both methods require keys to have been previously configured between the two routers.

With a CA, a router authenticates itself to the remote router by sending a certificate to the remote router and performing some public key cryptography. Each router must send its own unique certificate that was issued and validated by the CA. This process works because the certificate of each router encapsulates the public key of the router, each certificate is authenticated by the CA, and all participating routers recognize the CA as an authenticating authority. This scheme is called IKE with an RSA signature.

Your router can continue sending its own certificate for multiple IPSec sessions and to multiple IPSec peers until the certificate expires. When its certificate expires, the router administrator must obtain a new one from the CA.

When your router receives a certificate from a peer from another domain (with a different CA), the certificate revocation list (CRL) downloaded from the CA of the router does not include certificate information about the peer. Therefore, you should check the CRL published by the configured trustpoint with the Lightweight Directory Access Protocol (LDAP) URL to ensure that the certificate of the peer has not been revoked.

To query the CRL published by the configured trustpoint with the LDAP URL, use the **query url** command in trustpoint configuration mode.

CA Registration Authorities

Some CAs have a registration authority (RA) as part of their implementation. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is offline.

How to Implement CA Interoperability

This section contains the following procedures:

Configuring a Router Hostname and IP Domain Name

This task configures a router hostname and IP domain name.

You must configure the hostname and IP domain name of the router if they have not already been configured. The hostname and IP domain name are required because the router assigns a fully qualified domain name (FQDN) to the keys and certificates used by IPsec, and the FQDN is based on the hostname and IP domain name you assign to the router. For example, a certificate named `router20.example.com` is based on a router hostname of `router20` and a router IP domain name of `example.com`.

SUMMARY STEPS

1. **configure**
2. **hostname** *name*
3. **domain name** *domain-name*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: <code>RP/0/RP0/CPU0:router# configure</code>	Enters XR Config mode.
Step 2	hostname <i>name</i> Example: <code>RP/0/RP0/CPU0:router(config)# hostname myhost</code>	Configures the hostname of the router.
Step 3	domain name <i>domain-name</i> Example:	Configures the IP domain name of the router.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config)# domain name mydomain.com	
Step 4	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Generating an RSA Key Pair

This task generates an RSA key pair.

RSA key pairs are used to sign and encrypt IKE key management messages and are required before you can obtain a certificate for your router.

SUMMARY STEPS

1. **crypto key generate rsa [usage keys | general-keys] [keypair-label]**
2. **crypto key zeroize rsa [keypair-label]**
3. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>crypto key generate rsa [usage keys general-keys] [keypair-label]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key generate rsa general-keys</pre>	<p>Generates RSA key pairs.</p> <ul style="list-style-type: none"> • Use the usage keys keyword to specify special usage keys; use the general-keys keyword to specify general-purpose RSA keys. • The <i>keypair-label</i> argument is the RSA key pair label that names the RSA key pairs.
Step 2	<p>crypto key zeroize rsa [keypair-label]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key zeroize rsa key1</pre>	<p>(Optional) Deletes all RSAs from the router.</p> <ul style="list-style-type: none"> • Under certain circumstances, you may want to delete all RSA keys from your router. For example, if you believe the RSA keys were compromised in some way and should no longer be used, you should delete the keys.

	Command or Action	Purpose
		<ul style="list-style-type: none"> To remove a specific RSA key pair, use the <i>keypair-label</i> argument.
Step 3	show crypto key mypubkey rsa Example: RP/0/RP0/CPU0:router# show crypto key mypubkey rsa	(Optional) Displays the RSA public keys for your router.

Importing a Public Key to the Router

This task imports a public key to the router.

A public key is imported to the router to authenticate the user.

SUMMARY STEPS

1. **crypto key import authentication rsa [usage keys | general-keys] [keypair-label]**
2. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	crypto key import authentication rsa [usage keys general-keys] [keypair-label] Example: RP/0/RP0/CPU0:router# crypto key import authentication rsa general-keys	Generates RSA key pairs. <ul style="list-style-type: none"> Use the usage keys keyword to specify special usage keys; use the general-keys keyword to specify general-purpose RSA keys. The <i>keypair-label</i> argument is the RSA key pair label that names the RSA key pairs.
Step 2	show crypto key mypubkey rsa Example: RP/0/RP0/CPU0:router# show crypto key mypubkey rsa	(Optional) Displays the RSA public keys for your router.

Declaring a Certification Authority and Configuring a Trusted Point

This task declares a CA and configures a trusted point.

SUMMARY STEPS

1. **configure**
2. **crypto ca trustpoint ca-name**
3. **enrollment url CA-URL**

4. **query url LDAP-URL**
5. **enrollment retry period minutes**
6. **enrollment retry count number**
7. **rsa keypair keypair-label**
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	crypto ca trustpoint ca-name Example: <pre>RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca</pre>	Declares a CA. <ul style="list-style-type: none"> • Configures a trusted point with a selected name so that your router can verify certificates issued to peers. • Enters trustpoint configuration mode.
Step 3	enrollment url CA-URL Example: <pre>RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll</pre>	Specifies the URL of the CA. <ul style="list-style-type: none"> • The URL should include any nonstandard cgi-bin script location.
Step 4	query url LDAP-URL Example: <pre>RP/0/RP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com</pre>	(Optional) Specifies the location of the LDAP server if your CA system supports the LDAP protocol.
Step 5	enrollment retry period minutes Example: <pre>RP/0/RP0/CPU0:router(config-trustp)# enrollment retry period 2</pre>	(Optional) Specifies a retry period. <ul style="list-style-type: none"> • After requesting a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a period of time (the retry period) the router will send another certificate request. • Range is from 1 to 60 minutes. Default is 1 minute.
Step 6	enrollment retry count number Example: <pre>RP/0/RP0/CPU0:router(config-trustp)# enrollment retry count 10</pre>	(Optional) Specifies how many times the router continues to send unsuccessful certificate requests before giving up. <ul style="list-style-type: none"> • The range is from 1 to 100.
Step 7	rsa keypair keypair-label Example:	(Optional) Specifies a named RSA key pair generated using the crypto key generate rsa command for this trustpoint.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-trustp)# rsakeypair mykey	<ul style="list-style-type: none"> • Not setting this key pair means that the trustpoint uses the default RSA key in the current configuration.
Step 8	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Authenticating the CA

This task authenticates the CA to your router.

The router must authenticate the CA by obtaining the self-signed certificate of the CA, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate), manually authenticate the public key of the CA by contacting the CA administrator to compare the fingerprint of the CA certificate.

SUMMARY STEPS

1. **crypto ca authenticate ca-name**
2. show crypto ca certificates

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>crypto ca authenticate ca-name</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto ca authenticate myca</pre>	Authenticates the CA to your router by obtaining a CA certificate, which contains the public key for the CA.
Step 2	<p>show crypto ca certificates</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show crypto ca certificates</pre>	(Optional) Displays information about the CA certificate.

Requesting Your Own Certificates

This task requests certificates from the CA.

You must obtain a signed certificate from the CA for each of your router's RSA key pairs. If you generated general-purpose RSA keys, your router has only one RSA key pair and needs only one certificate. If you previously generated special usage RSA keys, your router has two RSA key pairs and needs two certificates.

SUMMARY STEPS

1. **crypto ca enroll ca-name**
2. **show crypto ca certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	crypto ca enroll ca-name Example: RP/0/RP0/CPU0:router# crypto ca enroll myca	Requests certificates for all of your RSA key pairs. <ul style="list-style-type: none"> • This command causes your router to request as many certificates as there are RSA key pairs, so you need only perform this command once, even if you have special usage RSA key pairs. • This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate needs to be revoked, so you must remember this password. • A certificate may be issued immediately or the router sends a certificate request every minute until the enrollment retry period is reached and a timeout occurs. If a timeout occurs, contact your system administrator to get your request approved, and then enter this command again.
Step 2	show crypto ca certificates Example: RP/0/RP0/CPU0:router# show crypto ca certificates	(Optional) Displays information about the CA certificate.

Configuring Certificate Enrollment Using Cut-and-Paste

This task declares the trustpoint certification authority (CA) that your router should use and configures that trustpoint CA for manual enrollment by using cut-and-paste.

SUMMARY STEPS

1. **configure**
2. **crypto ca trustpoint ca-name**
3. enrollment terminal
4. Use the **commit** or **end** command.
5. **crypto ca authenticate ca-name**
6. **crypto ca enroll ca-name**
7. **crypto ca import ca-name certificate**

8. show crypto ca certificates

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	crypto ca trustpoint <i>ca-name</i> Example: RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca RP/0//CPU0:router(config-trustp)#	Declares the CA that your router should use and enters trustpoint configuration mode. <ul style="list-style-type: none"> • Use the <i>ca-name</i> argument to specify the name of the CA.
Step 3	enrollment terminal Example: RP/0/RP0/CPU0:router(config-trustp)# enrollment terminal	Specifies manual cut-and-paste certificate enrollment.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 5	crypto ca authenticate <i>ca-name</i> Example: RP/0/RP0/CPU0:router# crypto ca authenticate myca	Authenticates the CA by obtaining the certificate of the CA. <ul style="list-style-type: none"> • Use the <i>ca-name</i> argument to specify the name of the CA. Use the same name that you entered in Step 2, on page 81.
Step 6	crypto ca enroll <i>ca-name</i> Example: RP/0/RP0/CPU0:router# crypto ca enroll myca	Obtains the certificates for your router from the CA. <ul style="list-style-type: none"> • Use the <i>ca-name</i> argument to specify the name of the CA. Use the same name that you entered in Step 2.
Step 7	crypto ca import <i>ca-name</i> certificate Example: RP/0/RP0/CPU0:router# crypto ca import myca certificate	Imports a certificate manually at the terminal. <ul style="list-style-type: none"> • Use the <i>ca-name</i> argument to specify the name of the CA. Use the same name that you entered in Step 2.

	Command or Action	Purpose
		Note You must enter the crypto ca import command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)
Step 8	show crypto ca certificates Example: RP/0/RP0/CPU0:router# show crypto ca certificates	Displays information about your certificate and the CA certificate.

Configuration Examples for Implementing Certification Authority Interoperability

This section provides the following configuration example:

Configuring Certification Authority Interoperability: Example

The following example shows how to configure CA interoperability.

Comments are included within the configuration to explain various commands.

```
configure
hostname myrouter
domain name mydomain.com
end
```

```
Uncommitted changes found, commit them? [yes]:yes
```

```
crypto key generate rsa mykey
```

```
The name for the keys will be:mykey
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keypair
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [1024]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]
```

```
show crypto key mypubkey rsa
```

```
Key label:mykey
Type      :RSA General purpose
Size      :1024
Created   :17:33:23 UTC Thu Sep 18 2003
Data      :
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CB8D86
BF6707AA FD7E4F08 A1F70080 B9E6016B 8128004C B477817B BCF35106 BC60B06E
```

```
07A417FD 7979D262 B35465A6 1D3B70D1 36ACAFBD 7F91D5A0 CFB0EE91 B9D52C69
7CAF89ED F66A6A58 89EEF776 A03916CB 3663FB17 B7DBEBF8 1C54AF7F 293F3004
C15B08A8 C6965F1E 289DD724 BD40AF59 E90E44D5 7D590000 5C4BEA9D B5020301
0001
```

! The following commands declare a CA and configure a trusted point.

```
configure
crypto ca trustpoint myca
enrollment url http://xyz-ultra5
enrollment retry count 25
enrollment retry period 2
rsakeypair mykey
end
```

Uncommitted changes found, commit them? [yes]:yes

! The following command authenticates the CA to your router.

```
crypto ca authenticate myca
```

```
Serial Number :01
Subject Name :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Issued By :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Validity Start :07:00:00 UTC Tue Aug 19 2003
Validity End :07:00:00 UTC Wed Aug 19 2020
Fingerprint:58 71 FB 94 55 65 D4 64 38 91 2B 00 61 E9 F8 05
Do you accept this certificate?? [yes/no]:yes
```

! The following command requests certificates for all of your RSA key pairs.

```
crypto ca enroll myca
```

```
% Start certificate enrollment ...
% Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
% For security reasons your password will not be saved in the configuration.
% Please make a note of it.
```

Password:

Re-enter Password:

```
Fingerprint: 17D8B38D ED2BDF2E DF8ADB7F A7DBE35A
```

! The following command displays information about your certificate and the CA certificate.

```
show crypto ca certificates
```

```
Trustpoint :myca
=====
CA certificate
Serial Number :01
Subject Name :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Issued By :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Validity Start :07:00:00 UTC Tue Aug 19 2003
Validity End :07:00:00 UTC Wed Aug 19 2020
Router certificate
Key usage :General Purpose
Status :Available
Serial Number :6E
Subject Name :
```

```

unstructuredName=myrouter.mydomain.com,o=Cisco Systems
Issued By      :
               cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Validity Start :21:43:14 UTC Mon Sep 22 2003
Validity End   :21:43:14 UTC Mon Sep 29 2003
CRL Distribution Point
               ldap://coax-u10.cisco.com/CN=Root coax-u10 Certificate Manager,O=Cisco Systems

```

Where to Go Next

After you have finished configuring CA interoperability, you should configure IKE, IPSec, and SSL. IPSec in the *Implementing IPSec Network Security on* module, and SSL in the *Implementing Secure Socket Layer on* module. These modules are located in *System Security Configuration Guide for Cisco NCS 6000 Series Routers* (this publication).

Additional References

The following sections provide references related to implementing certification authority interoperability.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Public Key Infrastructure Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i> .

Standards

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

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

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

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 4

Implementing Keychain Management

This module describes how to implement keychain management on. Keychain management is a common method of authentication to configure shared secrets on all entities that exchange secrets such as keys, before establishing trust with each other. Routing protocols and network management applications on Cisco IOS XR software often use authentication to enhance security while communicating with peers.

Feature History for Implementing Keychain Management

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Configuring Keychain Management, on page 87](#)
- [Restrictions for Implementing Keychain Management, on page 87](#)
- [Information About Implementing Keychain Management, on page 87](#)
- [How to Implement Keychain Management, on page 88](#)
- [Configuration Examples for Implementing Keychain Management, on page 96](#)
- [Additional References, on page 97](#)

Prerequisites for Configuring Keychain Management

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for Implementing Keychain Management

You must be aware that changing the system clock impacts the validity of the keys in the existing configuration.

Information About Implementing Keychain Management

The keychain by itself has no relevance; therefore, it must be used by an application that needs to communicate by using the keys (for authentication) with its peers. The keychain provides a secure mechanism to handle the keys and rollover based on the lifetime. Border Gateway Protocol (BGP), Open Shortest Path First (OSPF),

and Intermediate System-to-Intermediate System (IS-IS) use the keychain to implement a hitless key rollover for authentication. BGP uses TCP authentication, which enables the authentication option and sends the Message Authentication Code (MAC) based on the cryptographic algorithm configured for the keychain. For information about BGP, OSPF, and IS-IS keychain configurations, see

To implement keychain management, you must understand the concept of key lifetime, which is explained in the next section.

Lifetime of Key

If you are using keys as the security method, you must specify the lifetime for the keys and change the keys on a regular basis when they expire. To maintain stability, each party must be able to store and use more than one key for an application at the same time. A keychain is a sequence of keys that are collectively managed for authenticating the same peer, peer group, or both.

Keychain management groups a sequence of keys together under a keychain and associates each key in the keychain with a lifetime.



Note Any key that is configured without a lifetime is considered invalid; therefore, the key is rejected during configuration.

The lifetime of a key is defined by the following options:

- Start-time—Specifies the absolute time.
- End-time—Specifies the absolute time that is relative to the start-time or infinite time.

Each key definition within the keychain must specify a time interval for which that key is activated; for example, lifetime. Then, during a given key's lifetime, routing update packets are sent with this activated key. Keys cannot be used during time periods for which they are not activated. Therefore, we recommend that for a given keychain, key activation times overlap to avoid any period of time for which no key is activated. If a time period occurs during which no key is activated, neighbor authentication cannot occur; therefore, routing updates can fail.

Multiple keychains can be specified.

How to Implement Keychain Management

This section contains the following procedures:

Configuring a Keychain

This task configures a name for the keychain.

You can create or modify the name of the keychain.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*

3. Use the **commit** or **end** command.
4. **show key chain** *key-chain-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: <pre>RP/0/RP0/CPU0:router(config)# key chain isis-keys RP/0/RP0/CPU0:router(config-isis-keys)#</pre>	Creates a name for the keychain. Note Configuring only the keychain name without any key identifiers is considered a nonoperation. When you exit the configuration, the router does not prompt you to commit changes until you have configured the key identifier and at least one of the XR Config mode attributes or keychain-key configuration mode attributes (for example, lifetime or key string).
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 4	show key chain <i>key-chain-name</i> Example: <pre>RP/0/RP0/CPU0:router# show key chain isis-keys</pre>	(Optional) Displays the name of the keychain. Note The <i>key-chain-name</i> argument is optional. If you do not specify a name for the <i>key-chain-name</i> argument, all the keychains are displayed.

What to do next

After completing keychain configuration, see the [Configuring a Tolerance Specification to Accept Keys, on page 89](#) section.

Configuring a Tolerance Specification to Accept Keys

This task configures the tolerance specification to accept keys for a keychain to facilitate a hitless key rollover for applications, such as routing and management protocols.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **accept-tolerance** *value* [**infinite**]
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: RP/0//CPU0:router(config)# key chain isis-keys	Creates a name for the keychain.
Step 3	accept-tolerance <i>value</i> [infinite] Example: RP/0//CPU0:router(config-isis-keys)# accept-tolerance infinite	Configures a tolerance value to accept keys for the keychain. <ul style="list-style-type: none"> • Use the <i>value</i> argument to set the tolerance range in seconds. The range is from 1 to 8640000. • Use the infinite keyword to specify that the tolerance specification is infinite.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring a Key Identifier for the Keychain

This task configures a key identifier for the keychain.

You can create or modify the key for the keychain.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*

3. **key** *key-id*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: RP/0//CPU0:router(config)# key chain isis-keys	Creates a name for the keychain.
Step 3	key <i>key-id</i> Example: RP/0//CPU0:router(config-isis-keys)# key 8	Creates a key for the keychain. The key ID number is translated from decimal to hexadecimal to create the command mode subprompt. <ul style="list-style-type: none"> • Use the <i>key-id</i> argument as a 48-bit integer.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After configuring a key identifier for the keychain, see the [Configuring the Text for the Key String, on page 91](#) section.

Configuring the Text for the Key String

This task configures the text for the key string.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **key** *key-id*

4. **key-string** [**clear** | **password**] *key-string-text*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: RP/0//CPU0:router(config)# key chain isis-keys	Creates a name for the keychain.
Step 3	key <i>key-id</i> Example: RP/0//CPU0:router(config-isis-keys)# key 8 RP/0//CPU0:router(config-isis-keys-0x8)#	Creates a key for the keychain.
Step 4	key-string [clear password] <i>key-string-text</i> Example: RP/0//CPU0:router(config-isis-keys-0x8)# key-string password 8	Specifies the text string for the key. <ul style="list-style-type: none"> • Use the clear keyword to specify the key string in clear text form; use the password keyword to specify the key in encrypted form. • For a string to be a valid password, it must comply with the following rules: <ul style="list-style-type: none"> •
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

After configuring the text for the key string, see the [Configuring the Keys to Generate Authentication Digest for the Outbound Application Traffic](#), on page 94 section.

Determining the Valid Keys

This task determines the valid keys for local applications to authenticate the remote peers.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **key** *key-id*
4. **accept-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*]
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: RP/0/RP0/CPU0:router(config)# key chain isis-keys	Creates a a name for the keychain.
Step 3	key <i>key-id</i> Example: RP/0/RP0/CPU0:router(config-isis-keys)# key 8 RP/0/RP0/CPU0:router(config-isis-keys-0x8)#	Creates a key for the keychain.
Step 4	accept-lifetime <i>start-time</i> [duration <i>duration-value</i> infinite <i>end-time</i>] Example: RP/0/RP0/CPU0:router(config-isis-keys)#	(Optional) Specifies the validity of the key lifetime in terms of clock time.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring the Keys to Generate Authentication Digest for the Outbound Application Traffic

This task configures the keys to generate authentication digest for the outbound application traffic.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **key** *key-id*
4. **send-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*]
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	key chain <i>key-chain-name</i> Example: RP/0/RP0/CPU0:router(config)# key chain isis-keys	Creates a name for the keychain.
Step 3	key <i>key-id</i> Example: RP/0/RP0/CPU0:router(config-isis-keys)# key 8 RP/0/RP0/CPU0:router(config-isis-keys-0x8)#	Creates a key for the keychain.
Step 4	send-lifetime <i>start-time</i> [duration <i>duration-value</i> infinite <i>end-time</i>] Example: RP/0/RP0/CPU0:router(config-isis-keys)#	(Optional) Specifies the set time period during which an authentication key on a keychain is valid to be sent. You can specify the validity of the key lifetime in terms of clock time. In addition, you can specify a start-time value and one of the following values: <ul style="list-style-type: none"> • duration keyword (seconds) • infinite keyword • <i>end-time</i> argument If you intend to set lifetimes on keys, Network Time Protocol (NTP) or some other time synchronization method is recommended.

	Command or Action	Purpose
Step 5	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuring the Cryptographic Algorithm

This task allows the key chain configuration to accept the choice of the cryptographic algorithm.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **key** *key-id*
4. **cryptographic-algorithm** [HMAC-MD5 | HMAC-SHA1-12 | HMAC-SHA1-20 | MD5 | SHA-1]
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters XR Config mode.
Step 2	<p>key chain <i>key-chain-name</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# key chain isis-keys RP/0/RP0/CPU0:router(config-isis-keys)#</pre>	Creates a name for the keychain.
Step 3	<p>key <i>key-id</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-keys)# key 8 RP/0/RP0/CPU0:router(config-isis-keys-0x8)#</pre>	Creates a key for the keychain.
Step 4	<p>cryptographic-algorithm [HMAC-MD5 HMAC-SHA1-12 HMAC-SHA1-20 MD5 SHA-1]</p>	Specifies the choice of the cryptographic algorithm. You can choose from the following list of algorithms:

	Command or Action	Purpose
	<p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-keys-0x8)# cryptographic-algorithm MD5</pre>	<ul style="list-style-type: none"> • HMAC-MD5 • HMAC-SHA1-12 • HMAC-SHA1-20 • MD5 • SHA-1 <p>The routing protocols each support a different set of cryptographic algorithms:</p> <ul style="list-style-type: none"> • Border Gateway Protocol (BGP) supports only HMAC-MD5 and HMAC-SHA1-12. • Intermediate System-to-Intermediate System (IS-IS) supports HMAC-MD5, SHA-1, MD5. • Open Shortest Path First (OSPF) supports MD5, HMAC-MD5.
<p>Step 5</p>	<p>Use the commit or end command.</p>	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Configuration Examples for Implementing Keychain Management

This section provides the following configuration example:

Configuring Keychain Management: Example

The following example shows how to configure keychain management:

```
configure
key chain isis-keys
accept-tolerance infinite
key 8
key-string mykey9labcd
cryptographic-algorithm MD5
```

```

send-lifetime 1:00:00 june 29 2006 infinite
accept-lifetime 1:00:00 june 29 2006 infinite
end

Uncommitted changes found, commit them? [yes]: yes

show key chain isis-keys

Key-chain: isis-keys/ -

accept-tolerance -- infinite
Key 8 -- text "1104000E120B520005282820"
  cryptographic-algorithm -- MD5
  Send lifetime: 01:00:00, 29 Jun 2006 - Always valid [Valid now]
  Accept lifetime: 01:00:00, 29 Jun 2006 - Always valid [Valid now]

```

Additional References

The following sections provide references related to implementing keychain management.

Related Documents

Related Topic	Document Title
Keychain management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Keychain Management Commands in the System Security Command Reference for Cisco NCS 6000 Series Routers</i>

Standards

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

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 5

Implementing Lawful Intercept

Lawful intercept is the process by which law enforcement agencies conduct electronic surveillance of circuit and packet-mode communications, authorized by judicial or administrative order. Service providers worldwide are legally required to assist law enforcement agencies in conducting electronic surveillance in both circuit-switched and packet-mode networks.

Only authorized service provider personnel are permitted to process and configure lawfully authorized intercept orders. Network administrators and technicians are prohibited from obtaining knowledge of lawfully authorized intercept orders, or intercepts in progress. Error messages or program messages for intercepts installed in the router are not displayed on the console.

Lawful Intercept is not a part of the Cisco IOS XR software by default. You have to install it separately by installing and activating `ncs6k-li.pkg`.

Feature History for Implementing Lawful Intercept

Release	Modification
Release 5.2.1	This feature was introduced.
Release 6.6.1	Support for Layer 2 networks is added.

- [Prerequisites for Implementing Lawful Intercept](#), on page 99
- [Restrictions for Implementing Lawful Intercept](#), on page 101
- [Information About Lawful Intercept Implementation](#), on page 101
- [Intercepting IPv4 and IPv6 Packets](#), on page 104
- [High Availability for Lawful Intercept](#), on page 106
- [Installing Lawful Intercept \(LI\) Package](#), on page 107
- [How to Configure SNMPv3 Access for Lawful Intercept on the Cisco ASR 9000 Series Router](#), on page 109
- [Configuration Example for Inband Management Plane Feature Enablement](#), on page 114
- [Additional References](#), on page 115

Prerequisites for Implementing Lawful Intercept

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Lawful intercept implementation also requires that these prerequisites are met:

- **Provisioned router**—The router must be already provisioned. For more information, see .



Tip For the purpose of lawful intercept taps, provisioning a loopback interface has advantages over other interface types.

- **Understanding of SNMP Server commands in Cisco IOS XR software**—Simple Network Management Protocol, version 3 (SNMP v3), which is the basis for lawful intercept enablement, is configured using commands described in the module *SNMP Server Commands* in *System Management Command Reference for Cisco NCS 6000 Series Routers*. To implement lawful intercept, you must understand how the SNMP server functions. For this reason, carefully review the information described in the module *Implementing SNMP* in *System Management Configuration Guide for Cisco NCS 6000 Series Routers*.
- **Lawful intercept must be explicitly disabled**—It is automatically enabled on a provisioned router. However, you should not disable LI if there is an active tap in progress, because this deletes the tap.
- **Management plane configured to enable SNMPv3**—Allows the management plane to accept SNMP commands, so that the commands go to the interface (preferably, a loopback) on the router. This allows the mediation device (MD) to communicate with a physical interface.
- **VACM views enabled for SNMP server**—View-based access control model (VACM) views must be enabled on the router.
- **Provisioned MD**—For detailed information, see the vendor documentation associated with your MD. For a list of MD equipment suppliers preferred by Cisco, see http://www.cisco.com/en/US/tech/tk583/tk799/tsd_technology_support_protocol_home.html.
- **VoIP surveillance-specific requirements**
 - **Lawful-intercept-enabled call agent**—A lawful-intercept-enabled call agent must support interfaces for communications with the MD, for the target of interest to provide signaling information to the MD. The MD extracts source and destination IP addresses and Real-Time Protocol (RTP) port numbers from the Session Description Protocol (SDP) signaling information for the target of interest. It uses these to form an SNMPv3 SET, which is sent to the router acting as the content IAP to provision the intercept for the target of interest.

The MD uses the CISCO-TAP2-MIB to set up communications between the router acting as the content IAP, and the MD.

The MD uses the CISCO-IP-TAP-MIB to set up the filter for the IP addresses and port numbers to be intercepted and derived from the SDP.
 - Routers to be used for calls by the target number must be provisioned for this purpose through the MD.
 - The MD that has been provisioned with the target number to be intercepted.
- **Data session surveillance-specific requirements**
 - Routers to be used by the data target that have been provisioned for this purpose through the MD.
 - **The MD that has been provisioned with the user login ID, mac address of the user CPE device, or the DSLAM physical location ID**—The IP address is the binding that is most frequently used to identify the target in the network. However, alternative forms of information that uniquely identify

the target in the network might be used in some network architectures. Such alternatives include the MAC address and the acct-session-id.

- The MD can be located anywhere in the network but must be reachable from the content IAP router, which is being used to intercept the target. MD should be reachable ONLY from global routing table and NOT from VRF routing table.

Restrictions for Implementing Lawful Intercept

The following restrictions are applicable for Lawful Intercept:

- Lawful intercept does not provide support for these features on Cisco NCS 6000 Series Router:
 - IPv6 multicast tapping
 - Per interface tapping
 - Replicating a single tap to multiple MDs
 - Tapping L2 flows
 - RTP encapsulation
- Lawful intercept is applied only on IP traffic. If IP traffic is configured to leave an egress port as MPLS-encapsulated frames, then lawful intercept will not apply to the egress port, even if overlapping taps are configured.

Information About Lawful Intercept Implementation

Cisco lawful intercept is based on service-independent intercept (SII) architecture and SNMPv3 provisioning architecture. SNMPv3 addresses the requirements to authenticate data origin and ensure that the connection from the router to the MD is secure. This ensures that unauthorized parties cannot forge an intercept target.

Lawful intercept offers these capabilities:

- Voice-over IP (VoIP) and data session intercept provisioning from the MD using SNMPv3
- Delivery of intercepted VoIP and data session data to the MD
- SNMPv3 lawful intercept provisioning interface
- Lawful intercept MIB: CISCO-TAP2-MIB, version 2
- CISCO-IP-TAP-MIB manages the Cisco intercept feature for IP and is used along with CISCO-TAP2-MIB to intercept IP traffic.
- User datagram protocol (UDP) encapsulation to the MD
- Replication and forwarding of intercepted packets to the MD
- Voice-over IP (VoIP) call intercept, based on any rules configured for received packets.
- Voice-over IP (VoIP) intercept with LI-enabled call agent

- Data session call intercept based on IP address

Interception Mode

The lawful intercept has two interception modes:

- **Global LI:** The taps are installed on all the line cards in the ingress direction. With the global tap, the traffic for the target can be intercepted regardless of ingress point. Only the tap that has wild cards in the interface field is supported.
- **Interface LI:** Taps each packet that is entering or leaving an interface without any additional filters.

Overlapping Taps

Traffic interception can be configured for two inter-communicating intercepted hosts using overlapping taps.

For example, consider two taps, one configured for all traffic from source address A and another for all traffic going to destination address B. When a packet arrives with source address A and destination address B, the packet is tapped by TAP1 in ingress and TAP2 in egress, and copies will be generated and forwarded to both mediation devices. Overlapping taps can be enabled using **overlap-tap enable** command in Global configuration mode.

Provisioning for VoIP Calls

Lawful Intercept provisioning for VoIP occurs in these ways:

- Security and authentication occurs because users define this through SNMPv3.
- The MD provisions lawful intercept information using SNMPv3.
- Network management occurs through standard MIBs.

Call Interception

VoIP calls are intercepted in this manner:

- The MD uses configuration commands to configure the intercept on the call control entity.
- The call control entity sends intercept-related information about the target to the MD.
- The MD initiates call content intercept requests to the content IAP router or trunk gateway through SNMPv3.
- The content IAP router or trunk gateway intercepts the call content, replicates it, and sends it to the MD in Packet Cable Electronic Surveillance UDP format. Specifically, the original packet starting at the first byte of the IP header is prefixed with a four-byte CCCID supplied by the MD in TAP2-MIB. It is then put into a UDP frame with the destination address and port of the MD.
- After replicated VoIP packets are sent to the MD, the MD then forwards a copy to a law-enforcement-agency-owned collection function, using a recognized standard.

Provisioning for Data Sessions

Provisioning for data sessions occurs in a similar way to the way it does for lawful intercept for VoIP calls. (See [Provisioning for VoIP Calls](#), on page 102.)

Data Interception

Data are intercepted in this manner:

- If a lawful intercept-enabled authentication or accounting server is not available, a sniffer device can be used to detect the presence of the target in the network.
 - The MD uses configuration commands to configure the intercept on the sniffer.
 - The sniffer device sends intercept-related information about the target to the MD.
- The MD initiates communication content intercept requests to the content IAP router using SNMPv3.
- The content IAP router intercepts the communication content, replicates it, and sends it to the MD in UDP format.
- Intercepted data sessions are sent from the MD to the collection function of the law enforcement agency, using a supported delivery standard for lawful intercept.

Information About the MD

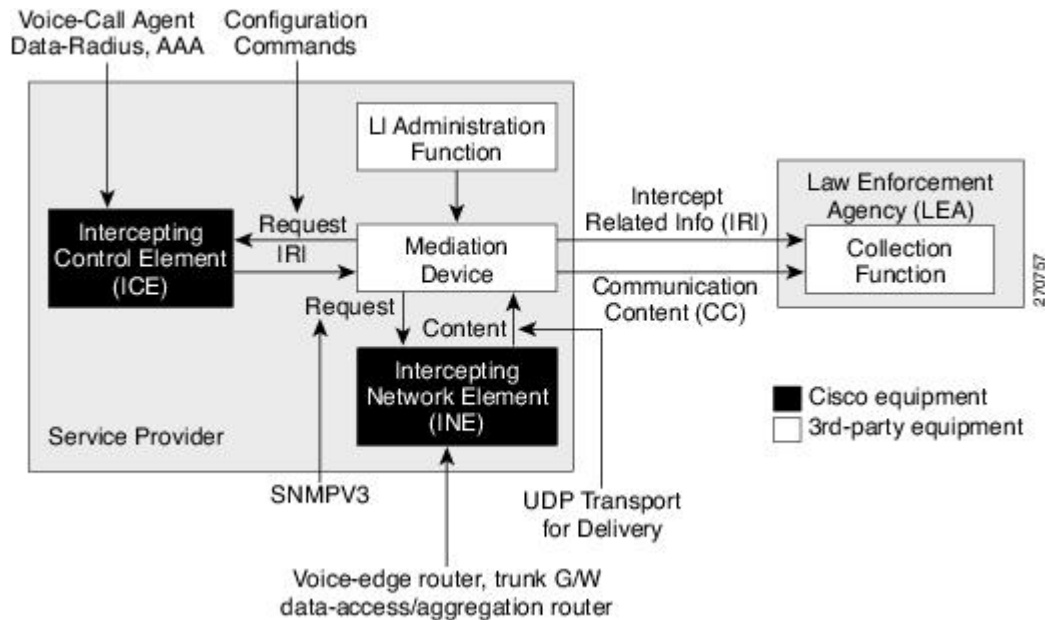
The MD performs these tasks:

- Activates the intercept at the authorized time and removes it when the authorized time period elapses.
- Periodically audits the elements in the network to ensure that:
 - *only* authorized intercepts are in place.
 - *all* authorized intercepts are in place.

Lawful Intercept Topology

This figure shows intercept access points and interfaces in a lawful intercept topology for both voice and data interception.

Figure 1: Lawful Intercept Topology for Both Voice and Data Interception



Intercepting IPv4 and IPv6 Packets

This section provides details for intercepting IPv4 and IPv6 packets supported on the Cisco NCS 6000 Series Router.

Lawful Intercept Filters

The filters used for classifying a tap are:

- IP address type
- Destination address
- Destination mask
- Source address
- Source mask
- ToS (Type of Service) and ToS mask
- Protocol
- Destination port with range
- Source port with range
- VRF (VPN Routing and Forwarding)
- Flow ID

Intercepting Packets Based on Flow ID (Applies to IPv6 only)

To further extend filtration criteria for IPv6 packets, an additional support to intercept IPv6 packets based on flow ID has been introduced on the Cisco NCS 6000 Series Router. All IPv6 packets are intercepted based on the fields in the IPv6 header which comprises numerous fields defined in IPv6 Header Field Details table:



Note The field length or payload length is not used for intercepting packets.

Table 4: IPv6 Header Field Details

IPv6 Field Name	Field Description	Field Length
Version	IPv6 version number.	4 bits
Traffic Class	Internet traffic priority delivery value.	8 bits
Flow ID (Flow Label)	Used for specifying special router handling from source to destination(s) for a sequence of packets.	20 bits
Payload Length	Specifies the length of the data in the packet. When cleared to zero, the option is a hop-by-hop Jumbo payload.	16 bits unassigned
Next Header	Specifies the next encapsulated protocol. The values are compatible with those specified for the IPv4 protocol field.	8 bits
Hop Limit	For each router that forwards the packet, the hop limit is decremented by 1. When the hop limit field reaches zero, the packet is discarded. This replaces the TTL field in the IPv4 header that was originally intended to be used as a time based hop limit.	8 bits unsigned
Source Address	The IPv6 address of the sending node.	16 bytes
Destination Address	The IPv6 address of the destination node.	16 bytes

The flow ID or flow label is a 20 bit field in the IPv6 packet header that is used to discriminate traffic flows. Each flow has a unique flow ID. The filtration criteria to intercept packets matching a particular flow ID is defined in the tap configuration file. From the line card, the intercepted mapped flow IDs are sent to the next hop, specified in the MD configuration file. The intercepted packets are replicated and sent to the MD from the line card.

Intercepting VRF (6VPE) and 6PE Packets

This section provides information about intercepting VRF aware packets and 6PE packets. Before describing how it works, a basic understanding of 6VPE networks is discussed.

The MPLS VPN model is a true peer VPN model. It enforces traffic separations by assigning unique VPN route forwarding (VRF) tables to each customer's VPN at the provider content IAP router. Thus, users in a specific VPN cannot view traffic outside their VPN.

Cisco NCS 6000 Series Router supports intercepting IPv6 packets of the specified VRF ID for 6VPE. To distinguish traffic on VPN, VRFs are defined containing a specific VRF ID. The filter criteria to tap a particular VRF ID is specified in the tap. IPv6 packets are intercepted with the VRF context on both scenarios: imposition (ip2mpls) and disposition (mpls2ip).

The 6PE packets carry IPv6 packets over VPN. The packets do not have a VRF ID. Only IP traffic is intercepted; no MPLS based intercepts are supported. The IPv6 traffic is intercepted at the content IAP of the MPLS cloud at imposition (ip2mpls) and at disposition (mpls2ip).

Intercepting IPv6 packets is also performed for ip2tag and tag2ip packets. Ip2tag packets are those which are converted from IPv6 to Tagging (IPv6 to MPLS), and tag2ip packets are those which are converted from Tagging to IPv6 (MPLS to IPv6) at the provider content IAP router.

Encapsulation Type Supported for Intercepted Packets

Intercepted packets mapping the tap are replicated, encapsulated, and then sent to the MD. IPv4 and IPv6 packets are encapsulated using UDP (User Datagram Protocol) encapsulation. The replicated packets are forwarded to MD using UDP as the content delivery protocol.

The intercepted packet gets a new UDP header and IPv4 header. Information for IPv4 header is derived from MD configuration. Apart from the IP and UDP headers, a 4 byte channel identifier (CCCID) is also inserted after the UDP header in the packet. After adding the MD encapsulation, if the packet size is above the MTU, the egress LC CPU fragments the packet. Moreover, there is a possibility that the packet tapped is already a fragment. Each tap is associated with only one MD. Cisco NCS 6000 Series Router does not support forwarding replicated packets to multiple MDs.



Note Encapsulation types, such as RTP and RTP-NOR, are not supported.

Per Tap Drop Counter Support

Cisco NCS 6000 Series Router line cards provide SNMP server as an interface to export each tap forwarded to MD packet and drop counts. Any intercepted packets that are dropped prior to getting forwarded to the MD due to policer action are counted and reported. The drops due to policer action are the only drops that are counted under per tap drop counters. If a lawful intercept filter is modified, the packet counts are reset to 0.

High Availability for Lawful Intercept

High availability for lawful intercept provides operational continuity of the TAP flows and provisioned MD tables to reduce loss of information due to route processor fail over (RPFO).

To achieve continuous interception of a stream, when RP fail over is detected; MDs are required to re-provision all the rows relating to CISCO-TAP2-MIB, CISCO-IP-TAP-MIB, and CISCO-USER-CONNECTION-TAP-MIB to synchronize database view across RP and MD.



Note The high availability for lawful intercept is enabled by default from Release 4.2.0 onwards.

Preserving TAP and MD Tables during RP Fail Over

At any point in time, MD has the responsibility to detect the loss of the taps via SNMP configuration process.

After RPFO is completed, MD should re-provision all the entries in the stream tables, MD tables, and IP taps with the same values they had before fail over. As long as an entry is re-provisioned in time, existing taps will continue to flow without any loss.

The following restrictions are listed for re-provisioning MD and tap tables with respect to behavior of SNMP operation on `citapStreamEntry`, `cTap2StreamEntry`, `cTap2MediationEntry` MIB objects:

- After RPFO, table rows that are not re-provisioned, shall return `NO_SUCH_INSTANCE` value as result of SNMP Get operation.
- Entire row in the table must be created in a single configuration step, with exactly same values as before RPFO, and with the `rowStatus` as `CreateAndGo`. Only exception is the `cTap2MediationTimeout` object, that should reflect valid future time.

Replay Timer

The replay timer is an internal timeout that provides enough time for MD to re-provision tap entries while maintaining existing tap flows. It resets and starts on the active RP when RPFO takes place. The replay timer is a factor of number of LI entries in router with a minimum value of 10 minutes.

After replay timeout, interception stops on taps that are not re-provisioned.



Note In case high availability is not required, MD waits for entries to age out after fail over. MD cannot change an entry before replay timer expiry. It can either reinstall taps as is, and then modify; or wait for it to age out.

Installing Lawful Intercept (LI) Package

As LI is not a part of the Cisco IOS XR image by default, you need to install it separately.

Installing and Activating the LI Package

The Package Installation Envelope (PIE) files, are installable software files with the `.pie` extension. PIE files are used to copy one or more software components onto the router. A PIE may contain a single component, a group of components (called a package), or a set of packages (called a composite package).

Use the **show install committed** command in EXEC mode to verify the committed software packages.

To install the Lawful Intercept (LI) package, you must install and activate the

For more information about installing PIEs, refer to *Upgrading and Managing Cisco IOS XR Software* section of the *System Management Configuration Guide for Cisco NCS 6000 Series Routers*.

SUMMARY STEPS

1. **admin**

2. **install add** *tftp://<IP address of tftp server>/<location of pie on server>*
3. **install activate** *device:package*
4. **install commit**
5. **exit**
6. **show install committed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	admin Example: RP/0/RP0/CPU0:router# admin	Enters System Admin EXEC mode.
Step 2	install add <i>tftp://<IP address of tftp server>/<location of pie on server></i> Example: RP/0/RP0/CPU0:router(admin)# install add tftp://172.201.11.140/auto/tftp-users1/	Copies the contents of a package installation envelope (PIE) file to a storage device.
Step 3	install activate <i>device:package</i> Example: RP/0/RP0/CPU0:router(admin)# install activate disk0:	Activates the respective package and adds more functionality to the existing software.
Step 4	install commit Example: RP/0/RP0/CPU0:router(admin)# install commit	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 5	exit Example: RP/0/RP0/CPU0:router(admin)# exit	Exits from the admin mode.
Step 6	show install committed Example: RP/0/RP0/CPU0:router# show install committed	Shows the list of the committed software packages.

Deactivating the LI PIE

To uninstall the Lawful Intercept package, deactivate as shown in the following steps:

SUMMARY STEPS

1. **admin**
2. **install deactivate** *device:package*
3. **install commit**

4. **install remove** *device:package*
5. **exit**
6. **show install committed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	admin Example: RP/0/RP0/CPU0:router# admin	Enters System Admin EXEC mode.
Step 2	install deactivate <i>device:package</i> Example: RP/0/RP0/CPU0:router(admin)# install deactivate disk0:	Activates the respective package and adds more functionality to the existing software.
Step 3	install commit Example: RP/0/RP0/CPU0:router(admin)# install commit	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 4	install remove <i>device:package</i> Example: RP/0/RP0/CPU0:router(admin)# install remove disk0:	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 5	exit Example: RP/0/RP0/CPU0:router(admin)# exit	Exits from the admin mode.
Step 6	show install committed Example: RP/0/RP0/CPU0:router# show install committed	Shows the list of the committed software packages.

How to Configure SNMPv3 Access for Lawful Intercept on the Cisco ASR 9000 Series Router

Perform these procedures in the order presented to configure SNMPv3 for the purpose of Lawful Intercept enablement:

Disabling SNMP-based Lawful Intercept

Lawful Intercept is enabled by default on the Cisco NCS 6000 Series Routers after installing and activating the **ncs6k-li.pkg**.

- To disable Lawful Intercept, enter the **lawful-intercept disable** command in global configuration mode.

- To re-enable it, use the **no** form of this command.

Disabling SNMP-based Lawful Intercept: Example

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# lawful-intercept disable
```



Note The **lawful-intercept disable** command is available on the router, only after installing and activating the **ncs6k-li.pkg**.

All SNMP-based taps are dropped when lawful intercept is disabled.

Configuring the Inband Management Plane Protection Feature

If MPP was not earlier configured to work with another protocol, then ensure that the MPP feature is also not configured to enable the SNMP server to communicate with the mediation device for lawful interception. In such cases, MPP must be configured specifically as an inband interface to allow SNMP commands to be accepted by the router, using a specified interface or all interfaces.



Note Ensure this task is performed, even if you have recently migrated to Cisco IOS XR Software from Cisco IOS, and you had MPP configured for a given protocol.

For lawful intercept, a loopback interface is often the choice for SNMP messages. If you choose this interface type, you must include it in your inband management configuration.

For a more detailed discussion of the inband management interface, see the [Inband Management Interface, on page 118](#).

Related Tasks

- [Configuring a Device for Management Plane Protection for an Inband Interface, on page 120](#)

Related Examples

- [Configuring the Inband Management Plane Protection Feature: Example, on page 114](#)

Enabling the Mediation Device to Intercept VoIP and Data Sessions

The following SNMP server configuration tasks enable the Cisco SII feature on a router running Cisco IOS XR Software by allowing the MD to intercept VoIP or data sessions.

SUMMARY STEPS

1. **configure**
2. **snmp-server view** *view-name* **ciscoTap2MIB included**

3. **snmp-server view** *view-name* **ciscoUserConnectionTapMIB included**
4. **snmp-server group** *group-name* **v3auth read** *view-name* **write** *view-name* **notify** *view-name*
5. **snmp-server host** *ip-address* **traps version 3 auth** *username* **udp-port** *port-number*
6. **snmp-server user** *mduser-id* *groupname* **v3 auth md5** *md-password*
7. Use the **commit** or **end** command.
8. **show snmp users**
9. **show snmp group**
10. **show snmp view**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	snmp-server view <i>view-name</i> ciscoTap2MIB included Example: RP/0//CPU0:router(config)# snmp-server view TapName ciscoTap2MIB included	Creates or modifies a view record and includes the CISCO-TAP2-MIB family in the view. The SNMP management objects in the CISCO-TAP2-MIB that controls lawful intercepts are included. This MIB is used by the mediation device to configure and run lawful intercepts on targets sending traffic through the router.
Step 3	snmp-server view <i>view-name</i> ciscoUserConnectionTapMIB included Example: RP/0//CPU0:router(config)# snmp-server view TapName ciscoUserConnectionTapMIB included	Creates or modifies a view record and includes the CISCO-USER-CONNECTION-TAP-MIB family, to manage the Cisco intercept feature for user connections. This MIB is used along with the CISCO-TAP2-MIB to intercept and filter user traffic.
Step 4	snmp-server group <i>group-name</i> v3auth read <i>view-name</i> write <i>view-name</i> notify <i>view-name</i> Example: RP/0//CPU0:router(config)# snmp-server group TapGroup v3 auth read TapView write TapView notify TapView	Configures a new SNMP group that maps SNMP users to SNMP views. This group must have read, write, and notify privileges for the SNMP view.
Step 5	snmp-server host <i>ip-address</i> traps version 3 auth <i>username</i> udp-port <i>port-number</i> Example: RP/0//CPU0:router(config)# snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.

	Command or Action	Purpose
Step 6	<p>snmp-server user <i>mduser-id groupname v3 auth md5 md-password</i></p> <p>Example:</p> <pre>RP/0//CPU0:router(config)# snmp-server mduser-id TapGroup v3 auth md5 mdpassword</pre>	<p>Configures the MD user as part of an SNMP group, using the v3 security model and the HMAC MD5 algorithm, which you associate with the MD password.</p> <ul style="list-style-type: none"> • The <i>mduser-id</i> and <i>mdpassword</i> must match that configured on MD. Alternatively, these values must match those in use on the router. • Passwords must be eight characters or longer to comply with SNMPv3 security minimums. • Minimum Lawful Intercept security level is auth; The noauth option will not work, as it indicates noAuthnoPriv security level. The Lawful Intercept security level must also match that of the MD. • Choices other than MD5 are available on the router, but the MD values must match. <p>Most MDs default to or support only MD5.</p>
Step 7	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 8	<p>show snmp users</p> <p>Example:</p> <pre>RP/0//CPU0:router# show snmp users</pre>	Displays information about each SNMP username in the SNMP user table.
Step 9	<p>show snmp group</p> <p>Example:</p> <pre>RP/0//CPU0:router# show snmp group</pre>	Displays information about each SNMP group on the network.
Step 10	<p>show snmp view</p> <p>Example:</p> <pre>RP/0//CPU0:router# show snmp view</pre>	Displays information about the configured views, including the associated MIB view family name, storage type, and status.

Adding MD and TAP Objects

To keep the MD row in active state, the following objects are mandatory:

- cTap2MediationDestAddressType
- cTap2MediationDestAddress
- cTap2MediationDestPort
- cTap2MediationSrcInterface
- cTap2MediationTimeout
- cTap2MediationTransport
- cTap2MediationStatus

SUMMARY STEPS

1. Add MD.
2. Add TAP.
3. Activate TAP.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>Add MD.</p> <p>Example:</p> <pre>setany -v3 <ip-address> <user> cTap2MediationDestAddressType.1 <ipv4/ipv6> cTap2MediationDestAddress.1 <"ip"> cTap2MediationDestPort.1 "1234" cTap2MediationSrcInterface.1 0 cTap2MediationTransport.1 udp cTap2MediationNotificationEnable.1 true cTap2MediationTimeout.1 '7 de 6 14 3 4 5 6 2d 1 2' cTap2MediationStatus.1 createAndGo cTap2MediationDestAddressType.1 = ipv4(1) cTap2MediationDestAddress.1 = 46 01 01 02 cTap2MediationDestPort.1 = 1234 cTap2MediationSrcInterface.1 = 0 cTap2MediationTransport.1 = udp(1) cTap2MediationNotificationEnable.1 = true(1) cTap2MediationTimeout.1 = 2014-Jun-20,03:04:05.6,-1:2 cTap2MediationStatus.1 = createAndGo(4)</pre>	<p>Creates an MD for mediation services.</p> <p>To delete a MD, run:</p> <pre>setany -v3 <ip-address> <user> cTap2MediationStatus.1 6 cTap2MediationStatus.1 = destroy(6)</pre>
Step 2	<p>Add TAP.</p> <p>Example:</p> <pre>setany -v3 <ip-address> <user> citapStreamInterface.1.1 0 citapStreamAddrType.1.1 <ipv4/ipv6> citapStreamSourceAddress.1.1 "5a 1 1 2"</pre>	<p>Creates a TAP for stream operation.</p> <p>To delete a TAP, run:</p> <pre>setany -v3 <ip-address> <user> citapStreamStatus.1.1 6 cTap2StreamStatus.1.1 6 citapStreamStatus.1.1 = destroy(6) cTap2StreamStatus.1.1 = destroy(6)</pre>

	Command or Action	Purpose
	<pre> citapStreamSourceLength.1.1 32 citapStreamStatus.1.1 citapStreamInterface.1.1 = 0 citapStreamAddrType.1.1 = ipv4(1) citapStreamSourceAddress.1.1 = 5a 01 01 02 citapStreamSourceLength.1.1 = 32 citapStreamStatus.1.1 = createAndGo(4) </pre>	
Step 3	<p>Activate TAP.</p> <p>Example:</p> <pre> setany -v3 <ip-address> <user> cTap2StreamType.1.1 ip cTap2StreamInterceptEnable.1.1 true cTap2StreamStatus.1.1 createAndGo cTap2StreamType.1.1 = ip(1) cTap2StreamInterceptEnable.1.1 = true(1) cTap2StreamStatus.1.1 = createAndGo(4) </pre> <p>Example:</p> <p>To add TAP for L2VPN networks</p> <pre> setany -v3 <ip-address> <user> citapStreamInterface.4.1200 1125 citapStreamStatus.4.1200 createAndGo </pre>	Activates the TAP for stream operation.

Configuration Example for Inband Management Plane Feature Enablement

This example illustrates how to enable the MPP feature, which is disabled by default, for the purpose of lawful intercept.

Configuring the Inband Management Plane Protection Feature: Example

You must specifically enable management activities, either globally or on a per-inband-port basis, using this procedure. To globally enable inbound MPP, use the keyword **all** with the **interface** command, rather than use a particular interface type and instance ID with it.

```

RP/0//CPU0:router# configure
RP/0//CPU0:router (config)# control-plane
RP/0//CPU0:router (config-ctrl)# management-plane
RP/0//CPU0:router (config-mpp)# inband
RP/0//CPU0:router (config-mpp-inband)# interface loopback0
RP/0//CPU0:router (config-mpp-inband-Loopback0)# allow snmp
RP/0//CPU0:router (config-mpp-inband-Loopback0)# commit
RP/0//CPU0:router (config-mpp-inband-Loopback0)# exit
RP/0//CPU0:router (config-mpp-inband)# exit

```

```

RP/0//CPU0:router(config-mpp)# exit
RP/0//CPU0:router(config-ctr)# exit
RP/0//CPU0:router(config)# exit
RP/0//CPU0:router# show mgmt-plane inband interface loopback0

Management Plane Protection - inband interface

interface - Loopback0
    snmp configured -
        All peers allowed
RP/0//CPU0:router(config)# commit

```

Additional References

These sections provide references related to implementing lawful intercept.

Related Documents

Related Topic	Document Title
Lawful Intercept commands	<i>System Security Command Reference for Cisco NCS 6000 Series Routers</i>
Implementing SNMP	<i>System Management Configuration Guide for Cisco NCS 6000 Series Routers</i>
SNMP Server commands	<i>System Management Command Reference for Cisco NCS 6000 Series Routers</i>

Standards

Standards	Title
A modular, open architecture designed for simple implementation that easily interacts with third-party equipment to meet service provider lawful intercept requirements.	See RFC-3924 under RFCs, on page 116 .
An application layer protocol that facilitates the exchange of management information between network devices. Part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite.	Simple Network Management Protocol Version 3 (SNMPv3)

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> • CISCO-TAP2-MIB, version 2 • CISCO-IP-TAP-MIB 	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC-3924	Cisco Architecture for Lawful Intercept in IP Networks

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access more content.	http://www.cisco.com/techsupport



CHAPTER 6

Implementing Management Plane Protection

The Management Plane Protection (MPP) feature in Cisco IOS XR software provides the capability to restrict the interfaces on which network management packets are allowed to enter a device. The MPP feature allows a network operator to designate one or more router interfaces as management interfaces.

Device management traffic may enter a device only through these management interfaces. After MPP is enabled, no interfaces except designated management interfaces accept network management traffic destined to the device. Restricting management packets to designated interfaces provides greater control over management of a device, providing more security for that device.

For information on MPP commands, see the *Management Plane Protection Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Implementing Management Plane Protection

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Implementing Management Plane Protection, on page 117](#)
- [Restrictions for Implementing Management Plane Protection, on page 117](#)
- [Information About Implementing Management Plane Protection, on page 118](#)
- [How to Configure a Device for Management Plane Protection, on page 120](#)
- [Configuration Examples for Implementing Management Plane Protection, on page 125](#)
- [Additional References, on page 127](#)

Prerequisites for Implementing Management Plane Protection

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for Implementing Management Plane Protection

The following restrictions are listed for implementing Management Plane Protection (MPP):

- Currently, MPP does not keep track of the denied or dropped protocol requests.

- MPP configuration does not enable the protocol services. MPP is responsible only for making the services available on different interfaces. The protocols are enabled explicitly.
- Management requests that are received on inband interfaces are not necessarily acknowledged there.
- Both Route Processor (RP) and distributed route processor (DRP) Ethernet interfaces are by default out-of-band interfaces and can be configured under MPP.
- The changes made for the MPP configuration do not affect the active sessions that are established before the changes.
- Currently, MPP controls only the incoming management requests for protocols, such as TFTP, Telnet, Simple Network Management Protocol (SNMP), Secure Shell (SSH), and HTTP.
- MPP does not support MIB.
- In a MPLS L3VPN, when MPP has VRF interface attached, it applies the VRF filter on an incoming interface through LPTS. When an incoming packet from the core interface has a different VRF, then MPP does not allow it.



Note When configuring a device for MPP for an inband interface the **Interface all** configuration does not apply specific VRF filter and allows traffic for all source and destination interfaces.

Information About Implementing Management Plane Protection

Before you enable the Management Plane Protection feature, you should understand the following concepts:

Inband Management Interface

An *inband management interface* is a Cisco IOS XR software physical or logical interface that processes management packets, as well as data-forwarding packets. An inband management interface is also called a *shared management interface*.

Out-of-Band Management Interface

Out-of-band refers to an interface that allows only management protocol traffic to be forwarded or processed. An *out-of-band management interface* is defined by the network operator to specifically receive network management traffic. The advantage is that forwarding (or customer) traffic cannot interfere with the management of the router, which significantly reduces the possibility of denial-of-service attacks.

Out-of-band interfaces forward traffic only between out-of-band interfaces or terminate management packets that are destined to the router. In addition, the out-of-band interfaces can participate in dynamic routing protocols. The service provider connects to the router's out-of-band interfaces and builds an independent overlay management network, with all the routing and policy tools that the router can provide.

Peer-Filtering on Interfaces

The peer-filtering option allows management traffic from specific peers, or a range of peers, to be configured.

Control Plane Protection Overview

A *control plane* is a collection of processes that run at the process level on a route processor and collectively provide high-level control for most Cisco IOS XR software functions. All traffic directly or indirectly destined to a router is handled by the control plane. Management Plane Protection operates within the Control Plane Infrastructure.

Management Plane

The *management plane* is the logical path of all traffic that is related to the management of a routing platform. One of three planes in a communication architecture that is structured in layers and planes, the management plane performs management functions for a network and coordinates functions among all the planes (management, control, and data). In addition, the management plane is used to manage a device through its connection to the network.

Examples of protocols processed in the management plane are Simple Network Management Protocol (SNMP), Telnet, HTTP, Secure HTTP (HTTPS), and SSH. These management protocols are used for monitoring and for command-line interface (CLI) access. Restricting access to devices to internal sources (trusted networks) is critical.

Management Plane Protection Feature

The MPP protection feature, as well as all the management protocols under MPP, are disabled by default. When you configure an interface as either out-of-band or inband, it automatically enables MPP. Consequently, this enablement extends to all the protocols under MPP.

If MPP is disabled and a protocol is activated, all interfaces can pass traffic.

When MPP is enabled with an activated protocol, the only default management interfaces allowing management traffic are the route processor (RP) and standby route processor (SRP) Ethernet interfaces. You must manually configure any other interface for which you want to enable MPP as a management interface, using the MPP CLI that follows. Afterwards, only the default management interfaces and those you have previously configured as MPP interfaces will accept network management packets destined for the device. All other interfaces drop such packets.



Note Logical interfaces (or any other interfaces not present on the data plane) filter packets based on the ingress physical interface.

After configuration, you can modify or delete a management interface.

Following are the management protocols that the MPP feature supports. These management protocols are also the only protocols affected when MPP is enabled.

- SSH, v1 and v2
- SNMP, all versions

- Telnet
- TFTP
- HTTP
- HTTPS

Benefits of the Management Plane Protection Feature

Implementing the MPP feature provides the following benefits:

- Greater access control for managing a device than allowing management protocols on all interfaces.
- Improved performance for data packets on non-management interfaces.
- Support for network scalability.
- Simplifies the task of using per-interface access control lists (ACLs) to restrict management access to the device.
- Fewer ACLs are needed to restrict access to the device.
- Prevention of packet floods on switching and routing interfaces from reaching the CPU.

How to Configure a Device for Management Plane Protection

This section contains the following tasks:

Configuring a Device for Management Plane Protection for an Inband Interface

Perform this task to configure a device that you have just added to your network or a device already operating in your network. This task shows how to configure MPP as an inband interface in which Telnet is allowed to access the router only through a specific interface.

Perform the following additional tasks to configure an inband MPP interface in non-default VRF.

- Configure the interface under the non-default inband VRF.
- Configure the global inband VRF.
- In the case of Telnet, configure the Telnet VRF server for the inband VRF.

SUMMARY STEPS

1. **configure**
2. control-plane
3. management-plane
4. inband
5. **interface** {*type instance* | **all**}
6. **allow** {*protocol* | **all**} [**peer**]
7. **address ipv4** {*peer-ip-address* | *peer ip-address/length*}

8. Use the **commit** or **end** command.
9. **show mgmt-plane [inband | out-of-band] [interface {type instance}]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	control-plane Example: RP/0/RP0/CPU0:router(config)# control-plane RP/0/RP0/CPU0:router(config-ctrl)#	Enters control plane configuration mode.
Step 3	management-plane Example: RP/0/RP0/CPU0:router(config-ctrl)# management-plane RP/0/RP0/CPU0:router(config-mpp)#	Configures management plane protection to allow and disallow protocols and enters management plane protection configuration mode.
Step 4	inband Example: RP/0/RP0/CPU0:router(config-mpp)# inband RP/0/RP0/CPU0:router(config-mpp-inband)#	Configures an inband interface and enters management plane protection inband configuration mode.
Step 5	interface {type instance all} Example: RP/0/RP0/CPU0:router(config-mpp-inband)# interface GigabitEthernet 0/6/0/1 RP/0/RP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)#	Configures a specific inband interface, or all inband interfaces. Use the interface command to enter management plane protection inband interface configuration mode. <ul style="list-style-type: none"> • Use the all keyword to configure all interfaces.
Step 6	allow {protocol all} [peer] Example: RP/0/RP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)# allow Telnet peer RP/0/RP0/CPU0:router(config-telnet-peer)#	Configures an interface as an inband interface for a specified protocol or all protocols. <ul style="list-style-type: none"> • Use the <i>protocol</i> argument to allow management protocols on the designated management interface. <ul style="list-style-type: none"> • HTTP or HTTPS • SNMP (also versions)

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Secure Shell (v1 and v2) • TFTP • Telnet <ul style="list-style-type: none"> • Use the all keyword to configure the interface to allow all the management traffic that is specified in the list of protocols. • (Optional) Use the peer keyword to configure the peer address on the interface.
Step 7	<p>address ipv4 {<i>peer-ip-address</i> <i>peer ip-address/length</i>}</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-telnet-peer)# address ipv4 10.1.0.0/16</pre>	<p>Configures the peer IPv4 address in which management traffic is allowed on the interface.</p> <ul style="list-style-type: none"> • Use the <i>peer-ip-address</i> argument to configure the peer IPv4 address in which management traffic is allowed on the interface. • Use the <i>peer ip-address/length</i> argument to configure the prefix of the peer IPv4 address.
Step 8	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 9	<p>show mgmt-plane [inband out-of-band] [interface {<i>type instance</i>}]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show mgmt-plane inband interface GigabitEthernet 0/6/0/1</pre>	<p>Displays information about the management plane, such as type of interface and protocols enabled on the interface.</p> <ul style="list-style-type: none"> • (Optional) Use the inband keyword to display the inband management interface configurations that are the interfaces that process management packets as well as data-forwarding packets. • (Optional) Use the out-of-band keyword to display the out-of-band interface configurations. • (Optional) Use the interface keyword to display the details for a specific interface.

Configuring a Device for Management Plane Protection for an Out-of-band Interface

Perform the following tasks to configure an out-of-band MPP interface.

- Configure the interface under the out-of-band VRF.
- Configure the global out-of-band VRF.
- In the case of Telnet, configure the Telnet VRF server for the out-of-band VRF.

SUMMARY STEPS

1. **configure**
2. control-plane
3. management-plane
4. out-of-band
5. **vrf** *vrf-name*
6. **interface** {*type instance* | **all**}
7. **allow** {*protocol* | **all**} [**peer**]
8. **address ipv6** {*peer-ip-address* | *peer ip-address/length*}
9. Use the **commit** or **end** command.
10. **show mgmt-plane** [**inband** | **out-of-band**] [**interface** {*type instance*} | **vrf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	control-plane Example: RP/0/RP0/CPU0:router(config)# control-plane RP/0/RP0/CPU0:router(config-ctrl)#	Enters control plane configuration mode.
Step 3	management-plane Example: RP/0/RP0/CPU0:router(config-ctrl)# management-plane RP/0/RP0/CPU0:router(config-mpp)#	Configures management plane protection to allow and disallow protocols and enters management plane protection configuration mode.

	Command or Action	Purpose
Step 4	<p>out-of-band</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpp)# out-of-band RP/0/RP0/CPU0:router(config-mpp-outband)#</pre>	Configures out-of-band interfaces or protocols and enters management plane protection out-of-band configuration mode.
Step 5	<p>vrf <i>vrf-name</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpp-outband)# vrf target</pre>	<p>Configures a Virtual Private Network (VPN) routing and forwarding (VRF) reference of an out-of-band interface.</p> <ul style="list-style-type: none"> Use the <i>vrf-name</i> argument to assign a name to a VRF.
Step 6	<p>interface {<i>type instance</i> all}</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpp-outband)# interface GigabitEthernet 0/6/0/2 RP/0/RP0/CPU0:router(config-mpp-outband-Gi0_6_0_2)#</pre>	<p>Configures a specific out-of-band interface, or all out-of-band interfaces, as an out-of-band interface. Use the interface command to enter management plane protection out-of-band configuration mode.</p> <ul style="list-style-type: none"> Use the all keyword to configure all interfaces.
Step 7	<p>allow {<i>protocol</i> all} [peer]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpp-outband-Gi0_6_0_2)# allow TFTP peer RP/0/RP0/CPU0:router(config-tftp-peer)#</pre>	<p>Configures an interface as an out-of-band interface for a specified protocol or all protocols.</p> <ul style="list-style-type: none"> Use the <i>protocol</i> argument to allow management protocols on the designated management interface. <ul style="list-style-type: none"> HTTP or HTTPS SNMP (also versions) Secure Shell (v1 and v2) TFTP Telnet Use the all keyword to configure the interface to allow all the management traffic that is specified in the list of protocols. (Optional) Use the peer keyword to configure the peer address on the interface.
Step 8	<p>address ipv6 {<i>peer-ip-address</i> <i>peer ip-address/length</i>}</p> <p>Example:</p>	Configures the peer IPv6 address in which management traffic is allowed on the interface.

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router(config-tftp-peer)# address ipv6 33::33</pre>	<ul style="list-style-type: none"> • Use the <i>peer-ip-address</i> argument to configure the peer IPv6 address in which management traffic is allowed on the interface. • Use the <i>peer ip-address/length</i> argument to configure the prefix of the peer IPv6 address.
Step 9	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 10	<p>show mgmt-plane [inband out-of-band] [interface {<i>type instance</i>} vrf]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show mgmt-plane out-of-band interface GigabitEthernet 0/6/0/2</pre>	<p>Displays information about the management plane, such as type of interface and protocols enabled on the interface.</p> <ul style="list-style-type: none"> • (Optional) Use the inband keyword to display the inband management interface configurations that are the interfaces that process management packets as well as data-forwarding packets. • (Optional) Use the out-of-band keyword to display the out-of-band interface configurations. • (Optional) Use the interface keyword to display the details for a specific interface. • (Optional) Use the vrf keyword to display the Virtual Private Network (VPN) routing and forwarding reference of an out-of-band interface.

Configuration Examples for Implementing Management Plane Protection

This section provides the following configuration example:

Configuring Management Plane Protection: Example

The following example shows how to configure inband and out-of-band interfaces for a specific IP address under MPP:

```

configure
control-plane
management-plane
inband
interface all
allow SSH
!
interface GigabitEthernet 0/6/0/0
allow all
allow SSH
allow Telnet peer
address ipv4 10.1.0.0/16
!
!
interface GigabitEthernet 0/6/0/1
allow Telnet peer
address ipv4 10.1.0.0/16
!
!
out-of-band
vrf my_out_of_band
interface GigabitEthernet 0/6/0/2
allow TFTP peer
address ipv6 33::33
!
!
!
!

show mgmt-plane

Management Plane Protection

inband interfaces
-----

interface - GigabitEthernet0_6_0_0
ssh configured -
    All peers allowed
telnet configured -
    peer v4 allowed - 10.1.0.0/16
all configured -
    All peers allowed
interface - GigabitEthernet0_6_0_1
telnet configured -
    peer v4 allowed - 10.1.0.0/16

interface - all
all configured -
    All peers allowed

outband interfaces
-----
interface - POS0_6_0_2
tftp configured -
    peer v6 allowed - 33::33

show mgmt-plane out-of-band vrf

Management Plane Protection -

```

```
out-of-band VRF - my_out_of_band
```

Additional References

The following sections provide references related to implementing management plane protection.

Related Documents

Related Topic	Document Title
MPP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Management Plane Protection Commands on System Monitoring Command Reference for Cisco NCS 6000 Series Routers</i>

Standards

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

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 7

Configuring Software Authentication Manager

Software Authentication Manager (SAM) is a component of the operating system that ensures that software being installed on the router is safe, and that the software does not run if its integrity has been compromised.

For information on SAM commands, see the *Software Authentication Manager Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

For information on setting the system clock, see the **clock set** command in *Clock Commands* module in *System Management Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Configuring Software Authentication Manager

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Configuring Software Authentication Manager, on page 129](#)
- [Information about Software Authentication Manager, on page 129](#)
- [How to set up a Prompt Interval for the Software Authentication Manager, on page 130](#)

Prerequisites for Configuring Software Authentication Manager

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information about Software Authentication Manager

For SAM to verify software during installation, the software to be installed must be in a Package for IOS/ENA (PIE) format. PIEs are digitally signed and SAM verifies the digital signature before allowing bits from that PIE to reside on the router. Each time an installed piece of software is run, SAM ensures that the integrity of the software is not been compromised since it was installed. SAM also verifies that software preinstalled on a flash card has not been tampered with while in transit.

When the initial image or a software package update is loaded on the router, SAM verifies the validity of the image by checking the expiration date of the certificate used to sign the image. If an error message is displayed

indicating that your certificate has expired, check the system clock and verify that it is accurate. If the system clock is not set correctly, the system does not function properly.

How to set up a Prompt Interval for the Software Authentication Manager

When the SAM detects an abnormal condition during boot time, it prompts the user to take action and waits for a certain interval. When the user does not respond within this interval, SAM proceeds with a predetermined action that can also be configured.

To set up the Prompt Interval, perform the following tasks.

SUMMARY STEPS

1. **configure**
2. **sam promptinterval** *time-interval* {**proceed** | **terminate**}
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	sam promptinterval <i>time-interval</i> { proceed terminate } Example: RP/0/RP0/CPU0:router(config)# sam prompt-interval 25 {proceed terminate}	Sets the prompt interval in seconds, after which the SAM either proceeds or terminates the interval. The Prompt interval ranges from 0 to 300 seconds. If the user responds, SAM considers it as a ‘Yes’ and proceeds with the next action. If the user does not respond, SAM considers it as a ‘No’ and terminates the action. The default time for which SAM waits is 10 seconds.
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.



CHAPTER 8

Implementing Secure Shell

Secure Shell (SSH) is an application and a protocol that provides a secure replacement to the Berkeley r-tools. The protocol secures sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley **rexec** and **rsh** tools.

Two versions of the SSH server are available: SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2). SSHv1 uses Rivest, Shamir, and Adelman (RSA) keys and SSHv2 uses either Digital Signature Algorithm (DSA) keys or Rivest, Shamir, and Adelman (RSA) keys. Cisco IOS XR software supports both SSHv1 and SSHv2.



Note Cisco IOS XR does not support X11 forwarding through an SSH connection.

This module describes how to implement Secure Shell on the .



Note For a complete description of the Secure Shell commands used in this , see the *Secure Shell Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Implementing Secure Shell

Release	Modification
Release 5.0.0	This feature was introduced.
Release 6.0	Support extended for the Netconf subsystem configuration to be vrf aware. At least one vrf needs to be configured to start the Netconf subsystem support. The configuration of the port is now optional.

- [Prerequisites for Implementing Secure Shell, on page 132](#)
- [Restrictions for Implementing Secure Shell, on page 132](#)
- [Information About Implementing Secure Shell, on page 133](#)
- [The Network Configuration Protocol, on page 137](#)
- [How to Implement Secure Shell, on page 139](#)
- [Enabling NETCONF over SSH, on page 144](#)
- [Configuration Examples for Implementing Secure Shell, on page 145](#)
- [Examples: Netconf over SSH, on page 146](#)

- [Additional References, on page 147](#)

Prerequisites for Implementing Secure Shell

The following prerequisites are required to implement Secure Shell:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Download the required image on your router. The SSH server and SSH client require you to have a crypto package (data encryption standard [DES], 3DES and AES) from Cisco downloaded on your router.
- To run an SSHv2 server, you must have a VRF. This may be the default VRF or a specific VRF. VRF changes are applicable only to the SSH v2 server.
- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA). For more information, see the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module in the *System Security Command Reference for Cisco NCS 6000 Series Routers* publication and *Configuring AAA Services on Cisco IOS XR Software* module in the *System Security Configuration Guide for Cisco NCS 6000 Series Routers* publication.
- AAA authentication and authorization must be configured correctly for Secure Shell File Transfer Protocol (SFTP) to work.

Restrictions for Implementing Secure Shell

The following are some basic SSH restrictions and limitations of the SFTP feature:

- A VRF is not accepted as inband if that VRF is already set as an out-of-band VRF. SSH v1 continues to bind only to the default VRF.
- In order for an outside client to connect to the router, the router needs to have an RSA (for SSHv1 or SSHv2) or DSA (for SSHv2) key pair configured. DSA and RSA keys are not required if you are initiating an SSH client connection from the router to an outside routing device. The same is true for SFTP: DSA and RSA keys are not required because SFTP operates only in client mode.
- In order for SFTP to work properly, the remote SSH server must enable the SFTP server functionality. For example, the SSHv2 server is configured to handle the SFTP subsystem with a line such as **/etc/ssh2/sshd2_config:**
- **subsystem-sftp /usr/local/sbin/sftp-server**
- The SFTP server is usually included as part of SSH packages from public domain and is turned on by default configuration.
- SFTP is compatible with sftp server version OpenSSH_2.9.9p2 or higher.
- RSA-based user authentication is supported in the SSH and SFTP servers. The support however, is not extended to the SSH client.
- Execution shell and SFTP are the only applications supported.

- The AES encryption algorithm is supported on the SSHv2 server and client, but not on the SSHv1 server and client. Any requests for an AES cipher sent by an SSHv2 client to an SSHv1 server are ignored, with the server using 3DES instead.
- The SFTP client does not support remote filenames containing wildcards (* ?, []). The user must issue the **sftp** command multiple times or list all of the source files from the remote host to download them on to the router. For uploading, the router SFTP client can support multiple files specified using a wildcard provided that the issues mentioned in the first through third bullets in this section are resolved.
- The cipher preference for the SSH server follows the order AES128, AES192, AES256, and, finally, 3DES. The server rejects any requests by the client for an unsupported cipher, and the SSH session does not proceed.
- Use of a terminal type other than vt100 is unsupported, and the software generates a warning message in this case.
- Password messages of “none” are unsupported on the SSH client.
- Because the router infrastructure does not provide support for UNIX-like file permissions, files created on the local device lose the original permission information. For files created on the remote file system, the file permission adheres to the umask on the destination host and the modification and last access times are the time of the copy.

Information About Implementing Secure Shell

To implement SSH, you should understand the following concepts:

SSH Server

The SSH server feature enables an SSH client to make a secure, encrypted connection to a Cisco router. This connection provides functionality that is similar to that of an inbound Telnet connection. Before SSH, security was limited to Telnet security. SSH allows a strong encryption to be used with the Cisco IOS XR software authentication. The SSH server in Cisco IOS XR software works with publicly and commercially available SSH clients.

SSH Client

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

The SSH client in the Cisco IOS XR software worked with publicly and commercially available SSH servers. The SSH client supported the ciphers of AES, 3DES, message digest algorithm 5 (MD5), SHA1, and password authentication. User authentication was performed in the Telnet session to the router. The user authentication mechanisms supported for SSH were RADIUS, TACACS+, and the use of locally stored usernames and passwords.

The SSH client supports setting DSCP value in the outgoing packets.

```
ssh client dscp <value from 0 - 63>
```

If not configured, the default DSCP value set in packets is 16 (for both client and server).

The SSH client supports the following options:

- **DSCP**—DSCP value for SSH client sessions.

```
RP/0/5/CPU0:router#configure
RP/0/5/CPU0:router(config)#ssh ?
  client  Provide SSH client service
  server  Provide SSH server service
  timeout Set timeout value for SSH
RP/0/5/CPU0:router(config)#ssh client ?
```

- **Knownhost**—Enable the host pubkey check by local database.
- **Source-interface**—Source interface for SSH client sessions.

```
RP/0/5/CPU0:router(config)#ssh client source-interface ?
ATM                ATM Network Interface(s)
BVI                Bridge-Group Virtual Interface
Bundle-Ether       Aggregated Ethernet interface(s)
Bundle-POS         Aggregated POS interface(s)
CEM                Circuit Emulation interface(s)
GigabitEthernet    GigabitEthernet/IEEE 802.3 interface(s)
IMA                ATM Network Interface(s)
IMtestmain         IM Test Interface
Loopback           Loopback interface(s)
MgmtEth            Ethernet/IEEE 802.3 interface(s)
Multilink          Multilink network interface(s)
Null               Null interface
PFItestmain        PFI Test Interface
PFItestnothw       PFI Test Not-HW Interface
POS                Packet over SONET/SDH network interface(s)
PW-Ether           PWHE Ethernet Interface
PW-IW              PWHE VC11 IP Interworking Interface
Serial             Serial network interface(s)
VASIleft           VASI Left interface(s)
VASIright          VASI Right interface(s)
test-bundle-channel Aggregated Test Bundle interface(s)
tunnel-ipsec       IPSec Tunnel interface(s)
tunnel-mte         MPLS Traffic Engineering P2MP Tunnel interface(s)
tunnel-te          MPLS Traffic Engineering Tunnel interface(s)
tunnel-tp          MPLS Transport Protocol Tunnel interface
RP/0/5/CPU0:router(config)#ssh client source-interface
RP/0/5/CPU0:router(config)#
```

- **VRF**—Source interface VRF for SSH client sessions:

```
RP/0/5/CPU0:router(config)#ssh client vrf ?
WORD  VRF name (max:32 chars)
RP/0/5/CPU0:router(config)#ssh client vrf shan ?
<cr>
RP/0/5/CPU0:router(config)#ssh client vrf shan
```

SSH also supports remote command execution as follows:

```
RP/0/5/CPU0:router#ssh ?
A.B.C.D  IPv4 (A.B.C.D) address
WORD     Hostname of the remote node
X:X::X   IPv6 (A:B:C:D...:D) address
vrf      vrf table for the route lookup
RP/0/5/CPU0:router#ssh 1.1.1.1 ?
cipher   Accept cipher type
command  Specify remote command (non-interactive)
source-interface Specify source interface
username Accept userid for authentication
```

```

<cr>
RP/0/5/CPU0:router#ssh 12.28.46.6 username admin command "show redundancy sum"
Password:

Wed Jan  9 07:05:27.997 PST
  Active Node      Standby Node
  -----
      0/4/CPU0      0/5/CPU0 (Node Ready, NSR: Not Configured)

RP/0/5/CPU0:router#

```

SFTP Feature Overview

SSH includes support for standard file transfer protocol (SFTP), a new standard file transfer protocol introduced in SSHv2. This feature provides a secure and authenticated method for copying router configuration or router image files.

The SFTP client functionality is provided as part of the SSH component and is always enabled on the router. Therefore, a user with the appropriate level can copy files to and from the router. Like the **copy** command, the **sftp** command can be used only in XR EXEC mode.

The SFTP client is VRF-aware, and you may configure the secure FTP client to use the VRF associated with a particular source interface during connections attempts. The SFTP client also supports interactive mode, where the user can log on to the server to perform specific tasks via the Unix server.

The SFTP Server is a sub-system of the SSH server. In other words, when an SSH server receives an SFTP server request, the SFTP API creates the SFTP server as a child process to the SSH server. A new SFTP server instance is created with each new request.

The SFTP requests for a new SFTP server in the following steps:

- The user runs the **sftp** command with the required arguments
- The SFTP API internally creates a child session that interacts with the SSH server
- The SSH server creates the SFTP server child process
- The SFTP server and client interact with each other in an encrypted format
- The SFTP transfer is subject to LPTS policer "SSH-Known". Low policer values will affect SFTP transfer speeds



Note In IOS-XR SW release 4.3.1 onwards the default policer value for SSH-Known has been reset from 2500pps to 300pps. Slower transfers are expected due to this change. You can adjust the lpts policer value for this punt cause to higher values that will allow faster transfers

When the SSH server establishes a new connection with the SSH client, the server daemon creates a new SSH server child process. The child server process builds a secure communications channel between the SSH client and server via key exchange and user authentication processes. If the SSH server receives a request for the sub-system to be an SFTP server, the SSH server daemon creates the SFTP server child process. For each incoming SFTP server subsystem request, a new SSH server child and a SFTP server instance is created. The SFTP server authenticates the user session and initiates a connection. It sets the environment for the client and the default directory for the user.

Once the initialization occurs, the SFTP server waits for the SSH_FXP_INIT message from the client, which is essential to start the file communication session. This message may then be followed by any message based on the client request. Here, the protocol adopts a 'request-response' model, where the client sends a request to the server; the server processes this request and sends a response.

The SFTP server displays the following responses:

- Status Response
- Handle Response
- Data Response
- Name Response



Note The server must be running in order to accept incoming SFTP connections.

RSA Based Host Authentication

Verifying the authenticity of a server is the first step to a secure SSH connection. This process is called the host authentication, and is conducted to ensure that a client connects to a valid server.

The host authentication is performed using the public key of a server. The server, during the key-exchange phase, provides its public key to the client. The client checks its database for known hosts of this server and the corresponding public-key. If the client fails to find the server's IP address, it displays a warning message to the user, offering an option to either save the public key or discard it. If the server's IP address is found, but the public-key does not match, the client closes the connection. If the public key is valid, the server is verified and a secure SSH connection is established.

The IOS XR SSH server and client had support for DSA based host authentication. But for compatibility with other products, like IOS, RSA based host authentication support is also added.

RSA Based User Authentication

One of the method for authenticating the user in SSH protocol is RSA public-key based user authentication. The possession of a private key serves as the authentication of the user. This method works by sending a signature created with a private key of the user. Each user has a RSA keypair on the client machine. The private key of the RSA keypair remains on the client machine.

The user generates an RSA public-private key pair on a unix client using a standard key generation mechanism such as ssh-keygen. The max length of the keys supported is 4096 bits, and the minimum length is 512 bits. The following example displays a typical key generation activity:

```
bash-2.05b$ ssh-keygen -b 1024 -t rsa
Generating RSA private key, 1024 bit long modulus
```

The public key must be in base64 encoded (binary) format for it to be imported correctly into the box. You can use third party tools available on the Internet to convert the key to the binary format.

Once the public key is imported to the router, the SSH client can choose to use the public key authentication method by specifying the request using the “-o” option in the SSH client. For example:

```
client$ ssh -o PreferredAuthentications=publickey 1.2.3.4
```

If a public key is not imported to a router using the RSA method, the SSH server initiates the password based authentication. If a public key is imported, the server proposes the use of both the methods. The SSH client then chooses to use either method to establish the connection. The system allows only 10 outgoing SSH client connections.

Currently, only SSH version 2 and SFTP server support the RSA based authentication. For more information on how to import the public key to the router, see the *Implementing Certification Authority Interoperability on* chapter in this guide.



Note The preferred method of authentication would be as stated in the SSH RFC. The RSA based authentication support is only for local authentication, and not for TACACS/RADIUS servers.

Authentication, Authorization, and Accounting (AAA) is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, see the *Authentication, Authorization, and Accounting Commands on* module in the *System Security Command Reference for Cisco NCS 6000 Series Routers* publication and the *Configuring AAA Services on* module in the *System Security Configuration Guide for Cisco NCS 6000 Series Routers* publication.

SSHv2 Client Keyboard-Interactive Authentication

An authentication method in which the authentication information is entered using a keyboard is known as keyboard-interactive authentication. This method is an interactive authentication method in the SSH protocol. This type of authentication allows the SSH client to support different methods of authentication without having to be aware of their underlying mechanisms.

Currently, the SSHv2 client supports the keyboard-interactive authentication. This type of authentication works only for interactive applications.



Note The password authentication is the default authentication method. The keyboard-interactive authentication method is selected if the server is configured to support only the keyboard-interactive authentication.

The Network Configuration Protocol

The Network Configuration Protocol (Netconf) provides mechanisms to install, manipulate, and delete the configuration of network devices. It uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. Yang is a data modeling language used with Netconf.

Netconf uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or application typically running as part of a network manager. The server is typically a network device.

The configuration of features need not be done the traditional way (using CLIs), the client application (controller) reads the Yang model and communicates with the Netconf server (IOS XR) accordingly.

Netconf Sessions and Operations

A Netconf session is the logical connection between a network configuration application and a network device. A device should be capable of supporting multiple sessions and atleast one Netconf session.

Characteristics of a netconf session:

- Netconf is connection-oriented - SSH is the underlying transport.
- The netconf client establishes session with the server.
- Netconf sessions are established with the *hello* message. Features and capabilities are announced.
- Sessions can be terminated using the *close* or *kill* messages.

Basic Netconf operations:

- Get configuration <get-config>
- Get all information <get>
- Edit configuration <edit-config>
- Copy configuration <copy-config>



Note <copy-config> does not support source attribute with “data store” at present.

- <lock>, <unlock>
- <kill-session>
- <close-session>
- Commit configuration <commit>

The Yang data model

Each feature has a defined Yang Model which is synthesized from the schemas. A model is published in a tree format and includes:

- Top level nodes and their subtrees
- Subtrees that augment nodes in other yang models

```
Example: The aaa Yang model
module: Cisco-IOS-XR-aaa-lib-cfg
  +--rw aaa
    +--rw accountings
      | +--rw accounting* [type listname]
      |   +--rw type                xr:Cisco-ios-xr-string
      |   +--rw listname            xr:Cisco-ios-xr-string
      |   +--rw rp-failover?        Aaa-accounting-rp-failover
      |   +--rw broadcast?         Aaa-accounting-broadcast
      |   +--rw type-xr?           Aaa-accounting
      |   +--rw method*            Aaa-method
      |   +--rw server-group-name* string
```

```

+--rw authorizations
| +--rw authorization* [type listname]
|   +--rw type                xr:Cisco-ios-xr-string
|   +--rw listname            xr:Cisco-ios-xr-string
|   +--rw method*             Aaa-method
|   +--rw server-group-name*  string
+--rw accounting-update!
| +--rw type                  Aaa-accounting-update
| +--rw periodic-interval?    uint32
+--rw authentications
  +--rw authentication* [type listname]
    +--rw type                xr:Cisco-ios-xr-string
    +--rw listname            xr:Cisco-ios-xr-string
    +--rw method*             Aaa-method
    +--rw server-group-name*  string

```

Advantages of using the Yang model are:

- Yang supports programmatic interfaces.
- Yang supports simplified network management applications.
- Yang supports interoperability that provides a standard way to model management data.

How to Implement Secure Shell

To configure SSH, perform the tasks described in the following sections:

Configuring SSH



Note For SSHv1 configuration, Step 1 to Step 4 are required. For SSHv2 configuration, Step 1 to Step 4 are optional.

SSH server supports setting DSCP value in the outgoing packets.

```
ssh server dscp <value from 0 - 63>
```

If not configured, the default DSCP value set in packets is 16 (for both client and server).

This is the syntax for setting DSCP value:

```
RP/0/5/CPU0:router(config)#ssh server dscp ?
<0-63> DSCP value range
```

```
RP/0/5/CPU0:router(config)#ssh server dscp 63 ?
<cr>
RP/0/5/CPU0:router(config)#ssh server dscp 63
RP/0/5/CPU0:router(config)#
```

```
RP/0/5/CPU0:router(config)#ssh client dscp ?
<0-63> DSCP value range
```

```
RP/0/5/CPU0:router(config)#ssh client dscp 0 ?
<cr>
RP/0/5/CPU0:router(config)#ssh client dscp 0
RP/0/5/CPU0:router(config)#
```

Perform this task to configure SSH.

SUMMARY STEPS

1. **configure**
2. **hostname** *hostname*
3. **domain name** *domain-name*
4. Use the **commit** or **end** command.
5. **crypto key generate rsa** [**usage keys** | **general-keys**] [*keypair-label*]
6. **crypto key generate dsa**
7. **configure**
8. **ssh timeout** *seconds*
9. Do one of the following:
 - **ssh server** [**vrf** *vrf-name* [**ipv4 access-list** IPv4 access-list name] [**ipv6 access-list** IPv6 access-list name]]
 - **ssh server v2**
10. Use the **commit** or **end** command.
11. **show ssh**
12. **show ssh session details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	hostname <i>hostname</i> Example: RP/0/RP0/CPU0:router(config)# hostname router1	Configures a hostname for your router.
Step 3	domain name <i>domain-name</i> Example: RP/0/RP0/CPU0:router(config)# domain name cisco.com	Defines a default domain name that the software uses to complete unqualified host names.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

	Command or Action	Purpose
Step 5	<p>crypto key generate rsa [usage keys general-keys] [keypair-label]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key generate rsa general-keys</pre>	<p>Generates an RSA key pair. The RSA key modulus can be in the range of 512 to 4096 bits.</p> <ul style="list-style-type: none"> To delete the RSA key pair, use the crypto key zeroize rsa command. This command is used for SSHv1 only.
Step 6	<p>crypto key generate dsa</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key generate dsa</pre>	<p>Enables the SSH server for local and remote authentication on the router. The supported key sizes are: 512, 768 and 1024 bits.</p> <ul style="list-style-type: none"> The recommended minimum modulus size is 1024 bits. Generates a DSA key pair. <p>To delete the DSA key pair, use the crypto key zeroize dsa command.</p> <ul style="list-style-type: none"> This command is used only for SSHv2.
Step 7	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	<p>Enters XR Config mode.</p>
Step 8	<p>ssh timeout <i>seconds</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh timeout 60</pre>	<p>(Optional) Configures the timeout value for user authentication to AAA.</p> <ul style="list-style-type: none"> If the user fails to authenticate itself to AAA within the configured time, the connection is terminated. If no value is configured, the default value of 30 seconds is used. The range is from 5 to 120.
Step 9	<p>Do one of the following:</p> <ul style="list-style-type: none"> ssh server [vrf <i>vrf-name</i> [ipv4 access-list [IPv4 access-list name] [ipv6 access-list [IPv6 access-list name]]] ssh server v2 <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh or RP/0/RP0/CPU0:router(config)# ssh server v2</pre>	<ul style="list-style-type: none"> (Optional) Brings up an SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed. <p>Note The SSH server can be configured for multiple VRF usage.</p>

	Command or Action	Purpose
		<ul style="list-style-type: none"> (Optional) Forces the SSH server to accept only SSHv2 clients if you configure the SSHv2 option by using the ssh server v2 command. If you choose the ssh server v2 command, only the SSH v2 client connections are accepted.
Step 10	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> Yes — Saves configuration changes and exits the configuration session. No —Exits the configuration session without committing the configuration changes. Cancel —Remains in the configuration session, without committing the configuration changes.
Step 11	<p>show ssh</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh</pre>	(Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router.
Step 12	<p>show ssh session details</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh session details</pre>	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Configuring the SSH Client

Perform this task to configure an SSH client.

SUMMARY STEPS

- configure**
- ssh client knownhost device : /filename**
- Use the **commit** or **end** command.
- ssh {ipv4-address | ipv6-address | hostname} [username user- cipher | source-interface type instance]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p>	Enters XR Config mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# configure	
Step 2	<p>ssh client knownhost <i>device</i> : /<i>filename</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey</pre>	<p>(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end.</p> <p>Note The complete path of the filename is required. The colon (:) and slash mark (/) are also required.</p>
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 4	<p>ssh {<i>ipv4-address</i> <i>ipv6-address</i> <i>hostname</i>} [username <i>user-</i> cipher source-interface <i>type instance</i>]</p>	<p>Enables an outbound SSH connection.</p> <ul style="list-style-type: none"> • To run an SSHv2 server, you must have a VRF. This may be the default or a specific VRF. VRF changes are applicable only to the SSH v2 server. • The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, the peer internally spawns an SSHv1 connection to the remote server. • The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only. • If the <i>hostname</i> argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.

- If you are using SSHv1 and your SSH connection is being rejected, the reason could be that the RSA key pair might have been zeroed out or that you have not successfully generated an RSA key pair for your router. Another reason could be that the SSH server to which the user is connecting to using SSHv1 client does not accept SSHv1 connections. Make sure that you have specified a hostname and domain. Then use the **crypto key generate rsa** command to generate an RSA host-key pair, and then enable the SSH server.
- If you are using SSHv2 and your SSH connection is being rejected, the reason could be that the DSA, RSA host-key pair might have been zeroed out. Make sure you follow similar steps as mentioned above to generate the required host-key pairs, and then enable the SSH server.
- When configuring the RSA or DSA key pair, you might encounter the following error messages:

- No hostname specified

You must configure a hostname for the router using the **hostname** command.

- No domain specified

You must configure a host domain for the router using the **domain-name** command.

- The number of allowable SSH connections is limited to the maximum number of virtual terminal lines configured for the router. Each SSH connection uses a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.



Note If you are using Putty version 0.63 or higher to connect to the SSH client, set the 'Chokes on PuTTYs SSH2 winadj request' option under SSH > Bugs in your Putty configuration to 'On.' This helps avoid a possible breakdown of the session whenever some long output is sent from IOS XR to the Putty client.

Enabling NETCONF over SSH

This task enables NETCONF over SSH. SSH is currently the only supported transport method .

If the client supports, Netconf over ssh can utilize the multi-channeling capabilities of IOS XR ssh server. For additional details about Multi-channeling in SSH, see *Implementing Secure Shell in System Security Configuration Guide*.

Prerequisites:

- k9sec pie must be installed, otherwise the port configuration for the netconf ssh server cannot be completed. (The Netconf subsystem for SSH, as well as, SSH cannot be configured without the k9sec pie.)
- Crypto keys must be generated prior to this configuration.
- The Netconf-YANG feature is packaged in the mgbl pie, which must be installed before enabling the Netconf-YANG agent.

SUMMARY STEPS

1. **configure**
2. **netconf-yang agent ssh**
3. **ssh server netconf** [*vrf vrf-name* [**ipv4 access-list** *ipv4 access list name*] [**ipv6 access-list** *ipv6 access list name*]]
4. **ssh server netconf port** *port-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters XR Config mode.
Step 2	netconf-yang agent ssh Example: RP/0/RP0/CPU0:router (config) # <code>netconf agent ssh</code>	Enables NETCONF agent over SSH connection. After NETCONF is enabled, the Yang model in the controller, can configure the relevant models. Note The Yang models can be retrieved from the router via NETCONF <get-schema> operation.
Step 3	ssh server netconf [<i>vrf vrf-name</i> [<i>ipv4 access-list</i> <i>ipv4 access list name</i>] [<i>ipv6 access-list ipv6 access list name</i>]] Example: RP/0/RP0/CPU0:router (config) # <code>ssh server netconf vrf netconfvrf ipv4 access-list InternetFilter</code>	Brings up the netconf subsystem support with SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the netconf subsystem of the ssh server before the port is opened. Note The netconf subsystem support with SSH server can be configured for use with multiple VRFs .
Step 4	ssh server netconf port <i>port-number</i> Example: RP/0/RP0/CPU0:router (config) # <code>ssh server netconf port 830</code>	Configures a port for the netconf ssh server. This command is optional. If no port is specified, port 830 is used by default. Note 830 is the IANA-assigned TCP port for NETCONF over SSH, but it can be changed using this command.

What to do next

The **show netconf-yang statistics** command and **show netconf-yang clients** command can be used to verify the configuration details of the netconf agent.

The **clear netconf-yang agent session** command clears the specified Netconf session (on the Netconf server side).

Configuration Examples for Implementing Secure Shell

This section provides the following configuration example:

Configuring Secure Shell: Example

This example shows how to configure SSHv2 by creating a hostname, defining a domain name, enabling the SSH server for local and remote authentication on the router by generating a DSA key pair, bringing up the SSH server, and saving the configuration commands to the running configuration file.

After SSH has been configured, the SFTP feature is available on the router.

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate dsa
configure
ssh server
end
```

Examples: Netconf over SSH

This section illustrates some examples relevant to Netconf:

Enabling netconf-yang for ssh transport and netconf subsystem for default vrf with default port (830)

```
config
netconf-yang agent ssh
ssh server netconf vrf default
!
!
```

Enabling netconf-yang for ssh transport and netconf subsystem for vrf *green* and vrf *red* with netconf port (831)

```
config
netconf-yang agent ssh
!
ssh server netconf vrf green
ssh server netconf vrf red
ssh server netconf port 831
!
!
```

Show command outputs

```
show netconf-yang statistics
Summary statistics
time per request| requests| total time| min time per request| max
other 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
close-session 0h 0m 0s 1ms| 4| 0h 0m 0s 3ms| 0h 0m 0s 0ms|
  0h 0m 0s 1ms| 0h 0m 0s 0ms|
kill-session 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get-schema 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get-config 0h 0m 0s 1ms| 1| 0h 0m 0s 1ms| 0h 0m 0s 1ms|
  0h 0m 0s 1ms| 0h 0m 0s 1ms|
edit-config 0h 0m 0s 2ms| 3| 0h 0m 0s 2ms| 0h 0m 0s 0ms|
```

```

0h 0m 0s 1ms|      0h 0m 0s 0ms|
commit            0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
cancel-commit    0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
lock            0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
unlock          0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
discard-changes 0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
validate        0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|

show netconf-yang clients
client session ID|  NC version|      client connect time|      last OP time|      last
OP type|      <lock>|
22969|          1.1|      0d 0h 0m 2s|      11:11:24|
close-session|      No|
15389|          1.1|      0d 0h 0m 1s|      11:11:25|      get-config|
          No|

```

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

Related Topic	Document Title
AAA commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Authentication, Authorization, and Accounting Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i> .
AAA configuration tasks	<i>Configuring AAA Services on</i> module in <i>System Security Configuration Guide for Cisco NCS 6000 Series Routers</i> .
Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Host Services and Applications Commands on</i> module in <i>IP Addresses and Services Command Reference for Cisco NCS 6000 Series Routers</i> .
IPSec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>IPSec Network Security Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i>
SSH commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Secure Shell Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i>

Standards

Standards	Title
Draft-ietf-secsh-userauth-17.txt	<i>SSH Authentication Protocol</i> , July 2003
Draft-ietf-secsh-connect-17.txt	<i>SSH Connection Protocol</i> , July 2003
Draft-ietf-secsh-architecture-14.txt	<i>SSH Protocol Architecture</i> , July 2003
Draft-ietf-secsh-transport-16.txt	<i>SSH Transport Layer Protocol</i> , July 2003

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 6020	Netconf/ Yang

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 9

Implementing Secure Socket Layer

This module describes how to implement SSL.

The Secure Socket Layer (SSL) protocol and Transport Layer Security (TLS) are application-level protocols that provide for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. SSL and TLS rely on certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Certificates are issued by certification authorities (CAs), such as VeriSign or Thawte. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date.

Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared quite freely, the private key is never given out. Each public-private key pair works together: Data encrypted with the public key can be decrypted only with the private key.



Note For a complete description of the Public Key Infrastructure (PKI) commands used here, see the *Public Key Infrastructure Commands* module in *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Feature History for Implementing Secure Socket Layer

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Implementing Secure Shell, on page 150](#)
- [Restrictions for Implementing Secure Shell, on page 150](#)
- [Information About Implementing Secure Shell, on page 151](#)
- [The Network Configuration Protocol, on page 155](#)
- [How to Implement Secure Shell, on page 157](#)
- [Enabling NETCONF over SSH, on page 162](#)
- [Configuration Examples for Implementing Secure Shell, on page 163](#)
- [Examples: Netconf over SSH, on page 164](#)
- [Additional References, on page 165](#)

Prerequisites for Implementing Secure Shell

The following prerequisites are required to implement Secure Shell:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Download the required image on your router. The SSH server and SSH client require you to have a crypto package (data encryption standard [DES], 3DES and AES) from Cisco downloaded on your router.
- To run an SSHv2 server, you must have a VRF. This may be the default VRF or a specific VRF. VRF changes are applicable only to the SSH v2 server.
- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA). For more information, see the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module in the *System Security Command Reference for Cisco NCS 6000 Series Routers* publication and *Configuring AAA Services on Cisco IOS XR Software* module in the *System Security Configuration Guide for Cisco NCS 6000 Series Routers* publication.
- AAA authentication and authorization must be configured correctly for Secure Shell File Transfer Protocol (SFTP) to work.

Restrictions for Implementing Secure Shell

The following are some basic SSH restrictions and limitations of the SFTP feature:

- A VRF is not accepted as inband if that VRF is already set as an out-of-band VRF. SSH v1 continues to bind only to the default VRF.
- In order for an outside client to connect to the router, the router needs to have an RSA (for SSHv1 or SSHv2) or DSA (for SSHv2) key pair configured. DSA and RSA keys are not required if you are initiating an SSH client connection from the router to an outside routing device. The same is true for SFTP: DSA and RSA keys are not required because SFTP operates only in client mode.
- In order for SFTP to work properly, the remote SSH server must enable the SFTP server functionality. For example, the SSHv2 server is configured to handle the SFTP subsystem with a line such as `/etc/ssh2/sshd2_config:`
- **subsystem-sftp /usr/local/sbin/sftp-server**
- The SFTP server is usually included as part of SSH packages from public domain and is turned on by default configuration.
- SFTP is compatible with sftp server version OpenSSH_2.9.9p2 or higher.
- RSA-based user authentication is supported in the SSH and SFTP servers. The support however, is not extended to the SSH client.
- Execution shell and SFTP are the only applications supported.

- The AES encryption algorithm is supported on the SSHv2 server and client, but not on the SSHv1 server and client. Any requests for an AES cipher sent by an SSHv2 client to an SSHv1 server are ignored, with the server using 3DES instead.
- The SFTP client does not support remote filenames containing wildcards (* ?, []). The user must issue the **sftp** command multiple times or list all of the source files from the remote host to download them on to the router. For uploading, the router SFTP client can support multiple files specified using a wildcard provided that the issues mentioned in the first through third bullets in this section are resolved.
- The cipher preference for the SSH server follows the order AES128, AES192, AES256, and, finally, 3DES. The server rejects any requests by the client for an unsupported cipher, and the SSH session does not proceed.
- Use of a terminal type other than vt100 is unsupported, and the software generates a warning message in this case.
- Password messages of “none” are unsupported on the SSH client.
- Because the router infrastructure does not provide support for UNIX-like file permissions, files created on the local device lose the original permission information. For files created on the remote file system, the file permission adheres to the umask on the destination host and the modification and last access times are the time of the copy.

Information About Implementing Secure Shell

To implement SSH, you should understand the following concepts:

SSH Server

The SSH server feature enables an SSH client to make a secure, encrypted connection to a Cisco router. This connection provides functionality that is similar to that of an inbound Telnet connection. Before SSH, security was limited to Telnet security. SSH allows a strong encryption to be used with the Cisco IOS XR software authentication. The SSH server in Cisco IOS XR software works with publicly and commercially available SSH clients.

SSH Client

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

The SSH client in the Cisco IOS XR software worked with publicly and commercially available SSH servers. The SSH client supported the ciphers of AES, 3DES, message digest algorithm 5 (MD5), SHA1, and password authentication. User authentication was performed in the Telnet session to the router. The user authentication mechanisms supported for SSH were RADIUS, TACACS+, and the use of locally stored usernames and passwords.

The SSH client supports setting DSCP value in the outgoing packets.

```
ssh client dscp <value from 0 - 63>
```

If not configured, the default DSCP value set in packets is 16 (for both client and server).

The SSH client supports the following options:

- **DSCP**—DSCP value for SSH client sessions.

```
RP/0/5/CPU0:router#configure
RP/0/5/CPU0:router(config)#ssh ?
  client  Provide SSH client service
  server  Provide SSH server service
  timeout Set timeout value for SSH
RP/0/5/CPU0:router(config)#ssh client ?
```

- **Knownhost**—Enable the host pubkey check by local database.
- **Source-interface**—Source interface for SSH client sessions.

```
RP/0/5/CPU0:router(config)#ssh client source-interface ?
ATM          ATM Network Interface(s)
BVI          Bridge-Group Virtual Interface
Bundle-Ether Aggregated Ethernet interface(s)
Bundle-POS   Aggregated POS interface(s)
CEM          Circuit Emulation interface(s)
GigabitEthernet GigabitEthernet/IEEE 802.3 interface(s)
IMA          ATM Network Interface(s)
IMtestmain   IM Test Interface
Loopback     Loopback interface(s)
MgmtEth      Ethernet/IEEE 802.3 interface(s)
Multilink    Multilink network interface(s)
Null         Null interface
PFItestmain  PFI Test Interface
PFItestnothw PFI Test Not-HW Interface
POS          Packet over SONET/SDH network interface(s)
PW-Ether     PWHE Ethernet Interface
PW-IW        PWHE VC11 IP Interworking Interface
Serial       Serial network interface(s)
VASILeft     VASI Left interface(s)
VASIRight    VASI Right interface(s)
test-bundle-channel Aggregated Test Bundle interface(s)
tunnel-ipsec IPsec Tunnel interface(s)
tunnel-mte   MPLS Traffic Engineering P2MP Tunnel interface(s)
tunnel-te    MPLS Traffic Engineering Tunnel interface(s)
tunnel-tp    MPLS Transport Protocol Tunnel interface
RP/0/5/CPU0:router(config)#ssh client source-interface
RP/0/5/CPU0:router(config)#
```

- **VRF**—Source interface VRF for SSH client sessions:

```
RP/0/5/CPU0:router(config)#ssh client vrf ?
WORD VRF name (max:32 chars)
RP/0/5/CPU0:router(config)#ssh client vrf shan ?
<cr>
RP/0/5/CPU0:router(config)#ssh client vrf shan
```

SSH also supports remote command execution as follows:

```
RP/0/5/CPU0:router#ssh ?
A.B.C.D IPv4 (A.B.C.D) address
WORD    Hostname of the remote node
X:X::X  IPv6 (A:B:C:D...:D) address
vrf     vrf table for the route lookup
RP/0/5/CPU0:router#ssh 1.1.1.1 ?
cipher      Accept cipher type
command     Specify remote command (non-interactive)
source-interface Specify source interface
username    Accept userid for authentication
```

```

<cr>
RP/0/5/CPU0:router#ssh 12.28.46.6 username admin command "show redundancy sum"
Password:

Wed Jan  9 07:05:27.997 PST
  Active Node      Standby Node
  -----
      0/4/CPU0      0/5/CPU0 (Node Ready, NSR: Not Configured)

RP/0/5/CPU0:router#

```

SFTP Feature Overview

SSH includes support for standard file transfer protocol (SFTP), a new standard file transfer protocol introduced in SSHv2. This feature provides a secure and authenticated method for copying router configuration or router image files.

The SFTP client functionality is provided as part of the SSH component and is always enabled on the router. Therefore, a user with the appropriate level can copy files to and from the router. Like the **copy** command, the **sftp** command can be used only in XR EXEC mode.

The SFTP client is VRF-aware, and you may configure the secure FTP client to use the VRF associated with a particular source interface during connections attempts. The SFTP client also supports interactive mode, where the user can log on to the server to perform specific tasks via the Unix server.

The SFTP Server is a sub-system of the SSH server. In other words, when an SSH server receives an SFTP server request, the SFTP API creates the SFTP server as a child process to the SSH server. A new SFTP server instance is created with each new request.

The SFTP requests for a new SFTP server in the following steps:

- The user runs the **sftp** command with the required arguments
- The SFTP API internally creates a child session that interacts with the SSH server
- The SSH server creates the SFTP server child process
- The SFTP server and client interact with each other in an encrypted format
- The SFTP transfer is subject to LPTS policer "SSH-Known". Low policer values will affect SFTP transfer speeds



Note In IOS-XR SW release 4.3.1 onwards the default policer value for SSH-Known has been reset from 2500pps to 300pps. Slower transfers are expected due to this change. You can adjust the lpts policer value for this punt cause to higher values that will allow faster transfers

When the SSH server establishes a new connection with the SSH client, the server daemon creates a new SSH server child process. The child server process builds a secure communications channel between the SSH client and server via key exchange and user authentication processes. If the SSH server receives a request for the sub-system to be an SFTP server, the SSH server daemon creates the SFTP server child process. For each incoming SFTP server subsystem request, a new SSH server child and a SFTP server instance is created. The SFTP server authenticates the user session and initiates a connection. It sets the environment for the client and the default directory for the user.

Once the initialization occurs, the SFTP server waits for the SSH_FXP_INIT message from the client, which is essential to start the file communication session. This message may then be followed by any message based on the client request. Here, the protocol adopts a 'request-response' model, where the client sends a request to the server; the server processes this request and sends a response.

The SFTP server displays the following responses:

- Status Response
- Handle Response
- Data Response
- Name Response



Note The server must be running in order to accept incoming SFTP connections.

RSA Based Host Authentication

Verifying the authenticity of a server is the first step to a secure SSH connection. This process is called the host authentication, and is conducted to ensure that a client connects to a valid server.

The host authentication is performed using the public key of a server. The server, during the key-exchange phase, provides its public key to the client. The client checks its database for known hosts of this server and the corresponding public-key. If the client fails to find the server's IP address, it displays a warning message to the user, offering an option to either save the public key or discard it. If the server's IP address is found, but the public-key does not match, the client closes the connection. If the public key is valid, the server is verified and a secure SSH connection is established.

The IOS XR SSH server and client had support for DSA based host authentication. But for compatibility with other products, like IOS, RSA based host authentication support is also added.

RSA Based User Authentication

One of the method for authenticating the user in SSH protocol is RSA public-key based user authentication. The possession of a private key serves as the authentication of the user. This method works by sending a signature created with a private key of the user. Each user has a RSA keypair on the client machine. The private key of the RSA keypair remains on the client machine.

The user generates an RSA public-private key pair on a unix client using a standard key generation mechanism such as ssh-keygen. The max length of the keys supported is 4096 bits, and the minimum length is 512 bits. The following example displays a typical key generation activity:

```
bash-2.05b$ ssh-keygen -b 1024 -t rsa
Generating RSA private key, 1024 bit long modulus
```

The public key must be in base64 encoded (binary) format for it to be imported correctly into the box. You can use third party tools available on the Internet to convert the key to the binary format.

Once the public key is imported to the router, the SSH client can choose to use the public key authentication method by specifying the request using the “-o” option in the SSH client. For example:

```
client$ ssh -o PreferredAuthentications=publickey 1.2.3.4
```

If a public key is not imported to a router using the RSA method, the SSH server initiates the password based authentication. If a public key is imported, the server proposes the use of both the methods. The SSH client then chooses to use either method to establish the connection. The system allows only 10 outgoing SSH client connections.

Currently, only SSH version 2 and SFTP server support the RSA based authentication. For more information on how to import the public key to the router, see the *Implementing Certification Authority Interoperability on* chapter in this guide.



Note The preferred method of authentication would be as stated in the SSH RFC. The RSA based authentication support is only for local authentication, and not for TACACS/RADIUS servers.

Authentication, Authorization, and Accounting (AAA) is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, see the *Authentication, Authorization, and Accounting Commands on* module in the *System Security Command Reference for Cisco NCS 6000 Series Routers* publication and the *Configuring AAA Services on* module in the *System Security Configuration Guide for Cisco NCS 6000 Series Routers* publication.

SSHv2 Client Keyboard-Interactive Authentication

An authentication method in which the authentication information is entered using a keyboard is known as keyboard-interactive authentication. This method is an interactive authentication method in the SSH protocol. This type of authentication allows the SSH client to support different methods of authentication without having to be aware of their underlying mechanisms.

Currently, the SSHv2 client supports the keyboard-interactive authentication. This type of authentication works only for interactive applications.



Note The password authentication is the default authentication method. The keyboard-interactive authentication method is selected if the server is configured to support only the keyboard-interactive authentication.

The Network Configuration Protocol

The Network Configuration Protocol (Netconf) provides mechanisms to install, manipulate, and delete the configuration of network devices. It uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. Yang is a data modeling language used with Netconf.

Netconf uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or application typically running as part of a network manager. The server is typically a network device.

The configuration of features need not be done the traditional way (using CLIs), the client application (controller) reads the Yang model and communicates with the Netconf server (IOS XR) accordingly.

Netconf Sessions and Operations

A Netconf session is the logical connection between a network configuration application and a network device. A device should be capable of supporting multiple sessions and atleast one Netconf session.

Characteristics of a netconf session:

- Netconf is connection-oriented - SSH is the underlying transport.
- The netconf client establishes session with the server.
- Netconf sessions are established with the *hello* message. Features and capabilities are announced.
- Sessions can be terminated using the *close* or *kill* messages.

Basic Netconf operations:

- Get configuration <get-config>
- Get all information <get>
- Edit configuration <edit-config>
- Copy configuration <copy-config>



Note <copy-config> does not support source attribute with “data store” at present.

- <lock>, <unlock>
- <kill-session>
- <close-session>
- Commit configuration <commit>

The Yang data model

Each feature has a defined Yang Model which is synthesized from the schemas. A model is published in a tree format and includes:

- Top level nodes and their subtrees
- Subtrees that augment nodes in other yang models

```
Example: The aaa Yang model
module: Cisco-IOS-XR-aaa-lib-cfg
  +--rw aaa
    +--rw accountings
      | +--rw accounting* [type listname]
      |   +--rw type                xr:Cisco-ios-xr-string
      |   +--rw listname            xr:Cisco-ios-xr-string
      |   +--rw rp-failover?       Aaa-accounting-rp-failover
      |   +--rw broadcast?        Aaa-accounting-broadcast
      |   +--rw type-xr?          Aaa-accounting
      |   +--rw method*           Aaa-method
      |   +--rw server-group-name* string
```



```

+--rw authorizations
| +--rw authorization* [type listname]
|   +--rw type                xr:Cisco-ios-xr-string
|   +--rw listname            xr:Cisco-ios-xr-string
|   +--rw method*             Aaa-method
|   +--rw server-group-name*  string
+--rw accounting-update!
| +--rw type                  Aaa-accounting-update
| +--rw periodic-interval?   uint32
+--rw authentications
   +--rw authentication* [type listname]
     +--rw type                xr:Cisco-ios-xr-string
     +--rw listname            xr:Cisco-ios-xr-string
     +--rw method*             Aaa-method
     +--rw server-group-name*  string

```

Advantages of using the Yang model are:

- Yang supports programmatic interfaces.
- Yang supports simplified network management applications.
- Yang supports interoperability that provides a standard way to model management data.

How to Implement Secure Shell

To configure SSH, perform the tasks described in the following sections:

Configuring SSH



Note For SSHv1 configuration, Step 1 to Step 4 are required. For SSHv2 configuration, Step 1 to Step 4 are optional.

SSH server supports setting DSCP value in the outgoing packets.

```
ssh server dscp <value from 0 - 63>
```

If not configured, the default DSCP value set in packets is 16 (for both client and server).

This is the syntax for setting DSCP value:

```
RP/0/5/CPU0:router(config)#ssh server dscp ?
<0-63> DSCP value range
```

```
RP/0/5/CPU0:router(config)#ssh server dscp 63 ?
<cr>
RP/0/5/CPU0:router(config)#ssh server dscp 63
RP/0/5/CPU0:router(config)#
```

```
RP/0/5/CPU0:router(config)#ssh client dscp ?
<0-63> DSCP value range
```

```
RP/0/5/CPU0:router(config)#ssh client dscp 0 ?
<cr>
RP/0/5/CPU0:router(config)#ssh client dscp 0
RP/0/5/CPU0:router(config)#
```

Perform this task to configure SSH.

SUMMARY STEPS

1. **configure**
2. **hostname** *hostname*
3. **domain name** *domain-name*
4. Use the **commit** or **end** command.
5. **crypto key generate rsa** [*usage keys* | *general-keys*] [*keypair-label*]
6. **crypto key generate dsa**
7. **configure**
8. **ssh timeout** *seconds*
9. Do one of the following:
 - **ssh server** [*vrf vrf-name* [*ipv4 access-list* IPv4 access-list name] [*ipv6 access-list* IPv6 access-list name]]
 - **ssh server v2**
10. Use the **commit** or **end** command.
11. **show ssh**
12. **show ssh session details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters XR Config mode.
Step 2	hostname <i>hostname</i> Example: RP/0/RP0/CPU0:router(config)# hostname router1	Configures a hostname for your router.
Step 3	domain name <i>domain-name</i> Example: RP/0/RP0/CPU0:router(config)# domain name cisco.com	Defines a default domain name that the software uses to complete unqualified host names.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

	Command or Action	Purpose
Step 5	<p>crypto key generate rsa [usage keys general-keys] [keypair-label]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key generate rsa general-keys</pre>	<p>Generates an RSA key pair. The RSA key modulus can be in the range of 512 to 4096 bits.</p> <ul style="list-style-type: none"> To delete the RSA key pair, use the crypto key zeroize rsa command. This command is used for SSHv1 only.
Step 6	<p>crypto key generate dsa</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# crypto key generate dsa</pre>	<p>Enables the SSH server for local and remote authentication on the router. The supported key sizes are: 512, 768 and 1024 bits.</p> <ul style="list-style-type: none"> The recommended minimum modulus size is 1024 bits. Generates a DSA key pair. <p>To delete the DSA key pair, use the crypto key zeroize dsa command.</p> <ul style="list-style-type: none"> This command is used only for SSHv2.
Step 7	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	<p>Enters XR Config mode.</p>
Step 8	<p>ssh timeout <i>seconds</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh timeout 60</pre>	<p>(Optional) Configures the timeout value for user authentication to AAA.</p> <ul style="list-style-type: none"> If the user fails to authenticate itself to AAA within the configured time, the connection is terminated. If no value is configured, the default value of 30 seconds is used. The range is from 5 to 120.
Step 9	<p>Do one of the following:</p> <ul style="list-style-type: none"> ssh server [vrf <i>vrf-name</i> [ipv4 access-list [IPv4 access-list name] [ipv6 access-list [IPv6 access-list name]]] ssh server v2 <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh or RP/0/RP0/CPU0:router(config)# ssh server v2</pre>	<ul style="list-style-type: none"> (Optional) Brings up an SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed. <p>Note The SSH server can be configured for multiple VRF usage.</p>

	Command or Action	Purpose
		<ul style="list-style-type: none"> • (Optional) Forces the SSH server to accept only SSHv2 clients if you configure the SSHv2 option by using the ssh server v2 command. If you choose the ssh server v2 command, only the SSH v2 client connections are accepted.
Step 10	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 11	<p>show ssh</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh</pre>	(Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router.
Step 12	<p>show ssh session details</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh session details</pre>	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Configuring the SSH Client

Perform this task to configure an SSH client.

SUMMARY STEPS

1. **configure**
2. **ssh client knownhost device : /filename**
3. Use the **commit** or **end** command.
4. **ssh {ipv4-address | ipv6-address | hostname} [username user- cipher | source-interface type instance]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p>	Enters XR Config mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# configure	
Step 2	<p>ssh client knownhost <i>device</i> : /<i>filename</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey</pre>	<p>(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end.</p> <p>Note The complete path of the filename is required. The colon (:) and slash mark (/) are also required.</p>
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 4	<p>ssh {<i>ipv4-address</i> <i>ipv6-address</i> <i>hostname</i>} [username <i>user-</i> cipher source-interface <i>type instance</i>]</p>	<p>Enables an outbound SSH connection.</p> <ul style="list-style-type: none"> • To run an SSHv2 server, you must have a VRF. This may be the default or a specific VRF. VRF changes are applicable only to the SSH v2 server. • The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, the peer internally spawns an SSHv1 connection to the remote server. • The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only. • If the <i>hostname</i> argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.

- If you are using SSHv1 and your SSH connection is being rejected, the reason could be that the RSA key pair might have been zeroed out or that you have not successfully generated an RSA key pair for your router. Another reason could be that the SSH server to which the user is connecting to using SSHv1 client does not accept SSHv1 connections. Make sure that you have specified a hostname and domain. Then use the **crypto key generate rsa** command to generate an RSA host-key pair, and then enable the SSH server.
- If you are using SSHv2 and your SSH connection is being rejected, the reason could be that the DSA, RSA host-key pair might have been zeroed out. Make sure you follow similar steps as mentioned above to generate the required host-key pairs, and then enable the SSH server.
- When configuring the RSA or DSA key pair, you might encounter the following error messages:

- No hostname specified

You must configure a hostname for the router using the **hostname** command.

- No domain specified

You must configure a host domain for the router using the **domain-name** command.

- The number of allowable SSH connections is limited to the maximum number of virtual terminal lines configured for the router. Each SSH connection uses a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.



Note If you are using Putty version 0.63 or higher to connect to the SSH client, set the 'Chokes on PuTTYs SSH2 winadj request' option under SSH > Bugs in your Putty configuration to 'On.' This helps avoid a possible breakdown of the session whenever some long output is sent from IOS XR to the Putty client.

Enabling NETCONF over SSH

This task enables NETCONF over SSH. SSH is currently the only supported transport method .

If the client supports, Netconf over ssh can utilize the multi-channeling capabilities of IOS XR ssh server. For additional details about Multi-channeling in SSH, see *Implementing Secure Shell in System Security Configuration Guide*.

Prerequisites:

- k9sec pie must be installed, otherwise the port configuration for the netconf ssh server cannot be completed. (The Netconf subsystem for SSH, as well as, SSH cannot be configured without the k9sec pie.)
- Crypto keys must be generated prior to this configuration.
- The Netconf-YANG feature is packaged in the mgbl pie, which must be installed before enabling the Netconf-YANG agent.

SUMMARY STEPS

1. **configure**
2. **netconf-yang agent ssh**
3. **ssh server netconf** [*vrf vrf-name* [**ipv4 access-list** *ipv4 access list name*] [**ipv6 access-list** *ipv6 access list name*]]
4. **ssh server netconf port** *port-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters XR Config mode.
Step 2	netconf-yang agent ssh Example: RP/0/RP0/CPU0:router (config) # <code>netconf agent ssh</code>	Enables NETCONF agent over SSH connection. After NETCONF is enabled, the Yang model in the controller, can configure the relevant models. Note The Yang models can be retrieved from the router via NETCONF <get-schema> operation.
Step 3	ssh server netconf [<i>vrf vrf-name</i> [<i>ipv4 access-list</i> <i>ipv4 access list name</i>] [<i>ipv6 access-list ipv6 access list name</i>]] Example: RP/0/RP0/CPU0:router (config) # <code>ssh server netconf vrf netconfvrf ipv4 access-list InternetFilter</code>	Brings up the netconf subsystem support with SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the netconf subsystem of the ssh server before the port is opened. Note The netconf subsystem support with SSH server can be configured for use with multiple VRFs .
Step 4	ssh server netconf port <i>port-number</i> Example: RP/0/RP0/CPU0:router (config) # <code>ssh server netconf port 830</code>	Configures a port for the netconf ssh server. This command is optional. If no port is specified, port 830 is used by default. Note 830 is the IANA-assigned TCP port for NETCONF over SSH, but it can be changed using this command.

What to do next

The **show netconf-yang statistics** command and **show netconf-yang clients** command can be used to verify the configuration details of the netconf agent.

The **clear netconf-yang agent session** command clears the specified Netconf session (on the Netconf server side).

Configuration Examples for Implementing Secure Shell

This section provides the following configuration example:

Configuring Secure Shell: Example

This example shows how to configure SSHv2 by creating a hostname, defining a domain name, enabling the SSH server for local and remote authentication on the router by generating a DSA key pair, bringing up the SSH server, and saving the configuration commands to the running configuration file.

After SSH has been configured, the SFTP feature is available on the router.

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate dsa
configure
ssh server
end
```

Examples: Netconf over SSH

This section illustrates some examples relevant to Netconf:

Enabling netconf-yang for ssh transport and netconf subsystem for default vrf with default port (830)

```
config
netconf-yang agent ssh
ssh server netconf vrf default
!
!
```

Enabling netconf-yang for ssh transport and netconf subsystem for vrf *green* and vrf *red* with netconf port (831)

```
config
netconf-yang agent ssh
!
ssh server netconf vrf green
ssh server netconf vrf red
ssh server netconf port 831
!
!
```

Show command outputs

```
show netconf-yang statistics
Summary statistics
time per request| requests| total time| min time per request| max
other 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
close-session 0h 0m 0s 1ms| 4| 0h 0m 0s 3ms| 0h 0m 0s 0ms|
  0h 0m 0s 1ms| 0h 0m 0s 0ms|
kill-session 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get-schema 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get 0h 0m 0s 0ms| 0| 0h 0m 0s 0ms| 0h 0m 0s 0ms|
  0h 0m 0s 0ms| 0h 0m 0s 0ms|
get-config 0h 0m 0s 1ms| 1| 0h 0m 0s 1ms| 0h 0m 0s 1ms|
  0h 0m 0s 1ms| 0h 0m 0s 1ms|
edit-config 0h 0m 0s 2ms| 3| 0h 0m 0s 2ms| 0h 0m 0s 0ms|
```



```

0h 0m 0s 1ms|      0h 0m 0s 0ms|
commit            0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
cancel-commit    0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
lock            0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
unlock          0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
discard-changes 0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
validate        0|      0h 0m 0s 0ms|      0h 0m 0s 0ms|
0h 0m 0s 0ms|      0h 0m 0s 0ms|      0h 0m 0s 0ms|

show netconf-yang clients
client session ID| NC version|      client connect time|      last OP time|      last
OP type|      <lock>|
22969|      1.1|      0d 0h 0m 2s|      11:11:24|
close-session|      No|
15389|      1.1|      0d 0h 0m 1s|      11:11:25|      get-config|
      No|

```

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

Related Topic	Document Title
AAA commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Authentication, Authorization, and Accounting Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i> .
AAA configuration tasks	<i>Configuring AAA Services on</i> module in <i>System Security Configuration Guide for Cisco NCS 6000 Series Routers</i> .
Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Host Services and Applications Commands on</i> module in <i>IP Addresses and Services Command Reference for Cisco NCS 6000 Series Routers</i> .
IPSec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>IPSec Network Security Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i>
SSH commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Secure Shell Commands on</i> module in <i>System Security Command Reference for Cisco NCS 6000 Series Routers</i>

Standards

Standards	Title
Draft-ietf-secsh-userauth-17.txt	<i>SSH Authentication Protocol</i> , July 2003
Draft-ietf-secsh-connect-17.txt	<i>SSH Connection Protocol</i> , July 2003
Draft-ietf-secsh-architecture-14.txt	<i>SSH Protocol Architecture</i> , July 2003
Draft-ietf-secsh-transport-16.txt	<i>SSH Transport Layer Protocol</i> , July 2003

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 6020	Netconf/ Yang

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport