CONTENTS

Preface xi

Changes to This Document xi

Obtaining Documentation and Submitting a Service Request xi

CHAPTER 1 Configuring AAA Services 1

Prerequisites for Configuring AAA Services 2

Restrictions for Configuring AAA Services 2

Information About Configuring AAA Services 2

User, User Groups, and Task Groups 2

User Categories 3

Root System Users 3

Root SDR Users 3

Secure Domain Router (SDR) Users 3

User Groups 4

Predefined User Groups 4

User-Defined User Groups 4

User Group Inheritance 4

Task Groups 5

Predefined Task Groups 5

User-Defined Task Groups 5

Group Inheritance 5

Cisco IOS XR Software Administrative Model 5

Administrative Access 6

AAA Database 6

Local Database 6

Remote Database 7

Remote AAA Configuration 7
Client Configuration 7
User Groups 7
Task Groups 7
AAA Configuration 7
Method Lists 8
Rollover Mechanism 8
Server Grouping 8
Authentication 8
Authentication of Root System User 8
Authentication of Non-Owner Secure Domain Router User 9
Authentication of Owner Secure Domain Router User 9
Authentication of Secure Domain Router User 9
Authentication Flow of Control 9
Korn Shell Authentication 10
Password Types 11
Task-Based Authorization 11
Task IDs 11
General Usage Guidelines for Task IDs 11
Task IDs for TACACS+ and RADIUS Authenticated Users 12
Task Maps 13
Format of the Task String 13
Privilege Level Mapping 15
XML Schema for AAA Services 15
About RADIUS 16
Network Security Situations in Which RADIUS is Unsuitable 17
RADIUS Operation 17
How to Configure AAA Services 18
Configuring Task Groups 18
Task Group Configuration 18
Configuring User Groups 20
Configuring Users 21
Configuring Router to RADIUS Server Communication 23
Configuring RADIUS Dead-Server Detection 26
Configuring Per VRF AAA 28
New Vendor-Specific Attributes (VSAs) 28
Configuring a TACACS+ Server 31
Configuring RADIUS Server Groups 33
Configuring TACACS+ Server Groups 35
Configuring AAA Method Lists 37
  Configuring Authentication Method Lists 37
    Authentication Configuration 37
    Creation of a Series of Authentication Methods 37
  Configuring Authorization Method Lists 39
    Authorization Configuration 40
    Creation of a Series of Authorization Methods 41
  Configuring Accounting Method Lists 43
    Accounting Configuration 43
    Creation of a Series of Accounting Methods 43
Generating Interim Accounting Records 45
Applying Method Lists for Applications 46
  Enabling AAA Authorization 46
    Method List Application 46
  Enabling Accounting Services 48
Configuring Login Parameters 49
Configuration Examples for Configuring AAA Services 50
  Configuring AAA Services: Example 50
Additional References 51

CHAPTER 2
Implementing Certification Authority Interoperability 53
  Prerequisites for Implementing Certification Authority 54
  Restrictions for Implementing Certification Authority 54
  Information About Implementing Certification Authority 54
    Supported Standards for Certification Authority Interoperability 54
  Certification Authorities 55
    Purpose of CAs 55
    IPSec Without CAs 56
    IPSec with CAs 56
    IPSec with Multiple Trustpoint CAs 56
    How IPSec Devices Use CA Certificates 57
    CA Registration Authorities 57
How to Implement CA Interoperability 57
  Configuring a Router Hostname and IP Domain Name 57
  Generating an RSA Key Pair 58
  Importing a Public Key to the Router 59
  Declaring a Certification Authority and Configuring a Trusted Point 60
  Authenticating the CA 62
  Requesting Your Own Certificates 62
  Configuring Certificate Enrollment Using Cut-and-Paste 63
Configuration Examples for Implementing Certification Authority Interoperability 65
  Configuring Certification Authority Interoperability: Example 65
Where to Go Next 67
Additional References 67

CHAPTER 3  Implementing Keychain Management 69
  Prerequisites for Configuring Keychain Management 69
  Restrictions for Implementing Keychain Management 69
  Information About Implementing Keychain Management 70
    Lifetime of a Key 70
  How to Implement Keychain Management 71
    Configuring a Keychain 71
    Configuring a Tolerance Specification to Accept Keys 72
    Configuring a Key Identifier for the Keychain 73
    Configuring the Text for the Key String 74
    Determining the Valid Keys 76
    Configuring the Keys to Generate Authentication Digest for the Outbound Application Traffic 77
    Configuring the Cryptographic Algorithm 79
Configuration Examples for Implementing Keychain Management 80
  Configuring Keychain Management: Example 80
Additional References 81

CHAPTER 4  Implementing Lawful Intercept 83
  Prerequisites for Implementing Lawful Intercept 83
  Restrictions for Implementing Lawful Intercept 85
  Information About Lawful Intercept Implementation 85
CHAPTER 5
Implementing Management Plane Protection 95
Prerequisites for Implementing Management Plane Protection 95
Restrictions for Implementing Management Plane Protection 96
Information About Implementing Management Plane Protection 96
Inband Management Interface 96
Out-of-Band Management Interface 96
Peer-Filtering on Interfaces 97
Control Plane Protection Overview 97
Management Plane 97
Management Plane Protection Feature 97
Benefits of the Management Plane Protection Feature 98
How to Configure a Device for Management Plane Protection 98
Configuring a Device for Management Plane Protection for an Inband Interface 98
Configuring a Device for Management Plane Protection for an Out-of-band Interface 101
Configuration Examples for Implementing Management Plane Protection 104
Configuring Management Plane Protection: Example 104
Additional References 105

CHAPTER 6
Configuring Software Authentication Manager 107
Prerequisites for Configuring Software Authentication Manager 107
Information about Software Authentication Manager 107
How to set up a Prompt Interval for the Software Authentication Manager 108

CHAPTER 7
Implementing Secure Shell 111
Prerequisites to Implementing Secure Shell 112
Restrictions for Implementing Secure Shell 112
Information About Implementing Secure Shell 113
SSH Server 113
SSH Client 113
SFTP Feature Overview 115
RSA Based Host Authentication 116
RSA Based User Authentication 116
SSHv2 Client Keyboard-Interactive Authentication 117
How to Implement Secure Shell 117
Configuring SSH 117
Configuring the SSH Client 121
Configuration Examples for Implementing Secure Shell 122
Configuring Secure Shell: Example 123
Additional References 123

CHAPTER 8
Implementing Secure Socket Layer 125
Prerequisites for Implementing Secure Socket Layer 126
Information About Implementing Secure Socket Layer 126
Purpose of Certification Authorities 126
How to Implement Secure Socket Layer 127
Configuring Secure Socket Layer 127
Configuration Examples for Implementing Secure Socket Layer 129
Configuring Secure Socket Layer: Example 129
Additional References 130

CHAPTER 9
Layer 2 Security Features 133
Security Features for Layer 2 VPLS Bridge Domains 133

CHAPTER 10
Implementing Traffic Storm Control under a VPLS Bridge 135
Prerequisites for Implementing Traffic Storm Control 135
Preface

This guide describes the configuration and examples for system security. For system security command descriptions, usage guidelines, task IDs, and examples, refer to the Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.

The preface contains the following sections:

- Changes to This Document, page xi
- Obtaining Documentation and Submitting a Service Request, page xi

Changes to This Document

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

Table 1: Changes to This Document

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>OL-24736-01</td>
<td>April 2011</td>
<td>Initial release of this document.</td>
</tr>
</tbody>
</table>

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What’s New in Cisco Product Documentation, at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html.

Subscribe to What’s New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.
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 Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

Feature History for Configuring AAA Services

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 4.1.0</td>
<td>Support for VRF aware TACACS+ was added.</td>
</tr>
</tbody>
</table>

- Prerequisites for Configuring AAA Services, page 2
- Restrictions for Configuring AAA Services, page 2
- Information About Configuring AAA Services, page 2
- How to Configure AAA Services, page 18
- Configuration Examples for Configuring AAA Services, page 50
- Additional References, page 51
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 12" 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 11 section)
User Categories

Router users are classified into the following categories:

- Root system user (complete administrative authority)
- 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 3 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 3 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 3 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 3 section.)

Secure Domain Router (SDR) Users

A SDR user has restricted access to an SDR as determined by the root-system user or 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 4 section.)
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 3 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-lr: Has the ability to control and monitor the specific secure domain router.

• root-system: Has the ability to control and monitor the entire system.

• 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).

The user group root-system has root system users as the only members. (See the Root System Users, on page 3 section.) The root-system user group has predefined authorization; that is, it has the complete responsibility for root-system user-managed resources and certain responsibilities in other SDRs.

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 re-inherited 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
- **root-system**: System-wide 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 4 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.

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.

The root-system user has the highest level of responsibility for the router. This user provisions secure domain routers and creates root SDR users. After being created, root SDR users take most of the responsibilities from the root-system user for the SDR. Root SDR users in turn can create secure domain router users. Root-system users and root SDR users have fixed permissions (task IDs) that cannot be changed by users.

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. A lockout of all root-system users is a serious issue that requires a system reload to recover the password.

- 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.

- Removing the flash card from disk0:, or a disk corruption, may deny auxiliary port authentication, which can affect certain system debugging abilities. However, if the console is available, the system is still accessible.

- Configuring command authorization or EXEC 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 EXEC 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 EXEC 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 12 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 and auxiliary 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 EXEC).

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 EXEC 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 Root System User

The root-system user can log in to any node in any secure domain router in the system. A user is a root-system user if he or she belongs to the root-system group. The root-system user may be defined in the local or remote AAA database.
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 37 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 or root-system 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 system user, root SDR user, or SDR user).

  • If the user has been configured as a member of a root-system user group, then AAA authenticates the user as a root-system 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 a root-system user group or 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 EXEC and external API clients can use this feature to optimize their operations. EXEC 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.
Korn Shell Authentication

The korn shell (ksh) is the primary shell for the auxiliary port of the route processor (RP), standby RP, and distributed RP cards and for console and auxiliary ports of line cards (LCs) and service processors (SPs). The following are some of the characteristics of ksh authentication:

- For security reasons, ksh authentication allows only root-system users who have a secret configured. A root-system user with a normal password will not be authenticated because the normal password is two-way encrypted and poses a security risk because the password information is stored in the flash disk, which can be easily decrypted.
- Every time a root-system user with a secret is configured using the normal AAA CLI, that user is a valid ksh user and no separate configuration is required.
- Ksh does not authenticate TACACS+ or RADIUS users, even if they are root-system users.
- Ksh authentication uses a single user password database, which means when a root-system user on a dSC is configured using the normal AAA CLI, that user can log in using this username password in any card. This includes the RP, standby RP, LC, and SP.
- Ksh authentication cannot be turned off or bypassed after the card is booted. To bypass authentication, a user needs a reload of the card. (See the "Bypassing ksh Authentication" section for details).
- The ksh run from the console (using the run command) is not authenticated because the run command needs the root-system task ID. Because the user is already root-system, the user is not authenticated again.

Bypassing ksh Authentication

Although the authentication to ksh is lightweight and depends on very few processes, there are cases when ksh authentication needs to be bypassed, including the following:

- dSC (Active RP) disk0 corruption
- Loss of Qnet connectivity
- Inability to determine the node ID of the dSC (Active RP)

To bypass ksh authentication, the user has to set the ROMMON variable AUX_AUTHEN_LEVEL to 0 and then reload the image. A reboot is required only on the card that has to bypass authentication.

The ROMMON variable AUX_AUTHEN_LEVEL can have one of the following values:

- 0—Authentication will be bypassed on the card.
- 1—Loose authentication. Authentication is performed on a best-effort basis and permits the user to access ksh if the system cannot access authentication information successfully.
- 2—Strict authentication. This is the default state.

Under no circumstances is authentication bypassed. Even if the authentication infrastructure is down, the system simply denies access.

For example, to bypass authentication on the card, enter the following:

```
rmon1> AUX_AUTHEN_LEVEL=0
rmon2> sync
rmon2> boot tftp:/ ...
```
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.

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Specifies a designation that permits only a read operation.</td>
</tr>
<tr>
<td>Write</td>
<td>Specifies a designation that permits a change operation and implicitly allows a read operation.</td>
</tr>
<tr>
<td>Execute</td>
<td>Specifies a designation that permits an access operation; for example ping and Telnet.</td>
</tr>
<tr>
<td>Debug</td>
<td>Specifies a designation that permits a debug operation.</td>
</tr>
</tbody>
</table>

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/RSP0/CPU0:router# show redundancy
```

Whereas, in administration EXEC mode, a user needs to be associated to both admin and system (read) task IDs and operations, as shown in the following example:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router (admin)# 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 13” 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 15” 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 EXEC mode to display all the tasks user1 can perform. For example:

**Example:**

```
Username:user1  
Password:  
RP/0/RSP0/CPU0:router# show user tasks  
```

```
Task: basic-services :READ  WRITE  EXECUTE  DEBUG
```

---
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/RSP0/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 15, the root-system user group is used; privilege level 14 maps to the user group owner-sdr.

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

```xml
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`.

With the RADIUS server, task IDs are defined using the Cisco-AVPair, as shown in the following example:

```xml
user = sampleuser2{
    member = bar
    Cisco-AVPair = "shell:tasks=$root-system,$cisco-support"
    Cisco-AVPair = "shell:priv-lvl=10"
}
```

### 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:
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:
   - ACCEPT—The user is authenticated.
   - REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

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

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and PPP, Serial Line Internet Protocol (SLIP), or EXEC 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 Step 4 for each task ID to be associated with the task group named in Step 2.
6. Use the commit or end command.
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example:</td>
<td>(in the context of a command)</td>
</tr>
<tr>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** | taskgroup  taskgroup-name |
| Example: | (in the context of a command) |
| Creates a name for a particular task group and enters task group configuration submode. |
| • Specific task groups can be removed from the system by specifying the no form of the taskgroup command. |

| **Step 3** | description  string |
| Example: | (in the context of a command) |
| (Optional) Creates a description of the task group named in Step 2. |

| **Step 4** | task  \{read | write | execute | debug\}  taskid-name |
| Example: | (in the context of a command) |
| Specifies a task ID to be associated with the task group named in Step 2. |
| • 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 Step 4 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: |
| • Yes—Saves configuration changes and exits the configuration session. |
| • No—Exits the configuration session without committing the configuration changes. |
| • Cancel—Remains in the configuration mode, 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 submode. Users can remove specific user groups by using the `no` form of the `usergroup` command. Deleting a user group 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 root-system and owner-sdr.

SUMMARY STEPS

1. configure
2. usergroup usergroup-name
3. description string
4. taskgroup taskgroup-name
5. Repeat Step 4, on page 23 for each task group to be associated with the user group named in Step Step 2, on page 20.
6. Use the `commit` or `end` command.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>usergroup usergroup-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# usergroup beta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Creates a name for a particular user group and enters user group configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# usergroup beta</td>
</tr>
<tr>
<td><strong>Notes:</strong></td>
<td>• Specific user groups can be removed from the system by specifying the <code>no</code> form of the <code>usergroup</code> command.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>description string</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ug)# description this is a sample user group</td>
</tr>
<tr>
<td></td>
<td>(Optional) Creates a description of the user group named in Step Step 2, on page 20.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>taskgroup taskgroup-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ug)# taskgroup beta</td>
</tr>
<tr>
<td></td>
<td>Associates the user group named in Step Step 2, on page 22 with the task group named in this step.</td>
</tr>
<tr>
<td></td>
<td>• The user group takes on the configuration attributes (task ID list and permissions) already defined for the entered task group.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Repeat Step Step 4, on page 23 for each task group to be associated with the user group named in Step Step 2, on page 20.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Use the commit or end command.</td>
</tr>
<tr>
<td></td>
<td>commit—Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>end—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>

**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 21 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

1. **configure**
   
2. **username** *user-name*
   
3. Do one of the following:
   - **password** {0 | 7} *password*
   - **secret** {0 | 5} *secret*

4. **group** *group-name*

5. Repeat Step 4, on page 23 for each user group to be associated with the user specified in Step 2, on page 22.

6. Use the **commit** or **end** command.

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>configure</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RX/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Creates a name for a new user (or identifies a current user) and enters username configuration submode.</td>
</tr>
<tr>
<td><strong>username</strong> <em>user-name</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RX/0/RSP0/CPU0:router(config)# username user1</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies a password for the user named in Step 2, on page 22.</td>
</tr>
<tr>
<td>Do one of the following:</td>
<td></td>
</tr>
<tr>
<td>- <strong>password</strong> {0</td>
<td>7} <em>password</em></td>
</tr>
<tr>
<td>- <strong>secret</strong> {0</td>
<td>5} <em>secret</em></td>
</tr>
<tr>
<td>Example:</td>
<td>RX/0/RSP0/CPU0:router(config-un)# password 0 pwd1 or RX/0/RSP0/CPU0:router(config-un)# secret 0 sec1</td>
</tr>
</tbody>
</table>
### Purpose

Assigns the user named in Step 2, on page 22 to a user group that has already been defined through the `usergroup` command.

- 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 4

**Example:**
```bash
RP/0/RSP0/CPU0:router(config-un)# group
sysadmin
```

Repeat Step 4, on page 23 for each user group to be associated with the user specified in Step 2, on page 22.

### 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:
  - **Yes**—Saves configuration changes and exits the configuration session.
  - **No**—Exits the configuration session without committing the configuration changes.
  - **Cancel**—Remains in the configuration mode, 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 23 or Configuring a TACACS+ Server, on page 31 section.)

### 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 25 through Step 6, on page 26 for each external server to be configured.
8. Use the commit or end command.
9. show radius
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Example: <code>RP/0/RSP0/CPU0:router# configure</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Specifies the hostname or IP address of the remote RADIUS server host.</th>
</tr>
</thead>
</table>
| `radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]` | - Use the `auth-port port-number` option to configure a specific UDP port on this RADIUS server to be used solely for authentication.  
- Use the `acct-port port-number` 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. |
| Example: `RP/0/RSP0/CPU0:router(config)# radius-server host host1` |

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Specifies the number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>radius-server retransmit retries</code></td>
<td>- In the example, the number of retransmission attempts is set to 5.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config)# radius-server retransmit 5</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Sets the number of seconds a router waits for a server host to reply before timing out.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>radius-server timeout seconds</code></td>
<td>- In the example, the interval timer is set to 10 seconds.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config)# radius-server timeout 10</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.</td>
</tr>
<tr>
<td>radius-server key {0 clear-text-key | 7 encrypted-key | clear-text-key}</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# radius-server key 0 samplekey</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Forces RADIUS to use the IP address of a specified interface or subinterface for all outgoing RADIUS packets.</td>
</tr>
<tr>
<td>radius source-interface \textit{type instance [vrf vrf-id]}</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# radius source-interface GigabitEthernet 0/3/0/1</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>The \textit{vrf} keyword enables the specification on a per-VRF basis.</td>
</tr>
<tr>
<td>Repeat <strong>Step 2, on page 25 through Step 6, on page 26</strong> for each external server to be configured.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td>commit—Saves the configuration changes and remains within the configuration session.</td>
<td></td>
</tr>
<tr>
<td>end—Prompts user to take one of these actions:</td>
<td></td>
</tr>
<tr>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
<td></td>
</tr>
<tr>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>(Optional) Displays information about the RADIUS servers that are configured in the system.</td>
</tr>
<tr>
<td>show radius</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# show radius</td>
</tr>
</tbody>
</table>

**What to Do Next**

After configuring router to RADIUS server communication, configure RADIUS server groups. (See the Configuring RADIUS Server Groups, on page 33 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-addr [auth-port auth-port] [acct-port acct-port]`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>radius-server deadtime minutes</code></td>
<td>Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config)# radius-server deadtime 5</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Establishes the time for the dead-criteria conditions for a RADIUS server to be marked as dead.</td>
<td></td>
</tr>
<tr>
<td><code>radius-server dead-criteria time seconds</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria time 5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Establishes the number of tries for the dead-criteria conditions for a RADIUS server to be marked as dead.</td>
<td></td>
</tr>
<tr>
<td><code>radius-server dead-criteria tries tries</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria tries 4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Use the <code>commit</code> or <code>end</code> command.</td>
<td></td>
</tr>
<tr>
<td><code>commit</code> — Saves the configuration changes and remains within the configuration session.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>end</code> — Prompts user to take one of these actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <code>Yes</code> — Saves configuration changes and exits the configuration session.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <code>No</code> — Exits the configuration session without committing the configuration changes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <code>Cancel</code> — Remains in the configuration mode, without committing the configuration changes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Displays dead-server-detection information that has been requested for a RADIUS server at the specified IP address.</td>
<td></td>
</tr>
<tr>
<td><code>show radius dead-criteria host ip-addr [auth-port auth-port] [acct-port acct-port]</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show radius dead-criteria host 172.19.192.80</td>
<td></td>
</tr>
</tbody>
</table>

## 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**

<table>
<thead>
<tr>
<th>VSA Name</th>
<th>Value Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td></td>
<td>The RADIUS VSAs—rad-serv, rad-serv-source-if, and rad-serv-vrf—must have the prefix &quot;aaa: &quot; before the VSA name.</td>
</tr>
<tr>
<td>rad-serv</td>
<td>string</td>
<td>Indicates the IP address, key, timeout, and retransmit number of a server and the group of the server. The VSA syntax follows: rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W]. 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. The key cannot contain any spaces; for &quot;retransmit V.&quot; &quot;V&quot; can range from 1 to 100; for &quot;timeout W,&quot; the &quot;W&quot; can range from 1 to 1000.</td>
</tr>
<tr>
<td>rad-serv-vrf</td>
<td>string</td>
<td>Specifies the name of the VRF that is used to transmit RADIUS packets. The VRF name matches the name that was specified through the <code>vrf</code> command.</td>
</tr>
</tbody>
</table>

This task configures RADIUS server groups per VRF. For information about configuring TACACS+ server groups per VRF, refer **Configuring TACACS+ Server Groups**, on page 35.
SUMMARY STEPS

1. configure
2. aaa group server radius group-name
3. server-private {hostname | ip-address} [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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> aaa group server radius group-name</td>
<td>Groups different server hosts into distinct lists and enters the server group configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# aaa group server radius radgroup1 RP/0/RSP0/CPU0:router(config-sg-radius)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> server-private {hostname</td>
<td>ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 timeout 5 RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 retransmit 3</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 4</strong> vrf vrf-name</td>
<td>Configures the VRF reference of an AAA RADIUS server group.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-sg-radius)# vrf v2.44.com</td>
<td>Note: Private server IP addresses can overlap with those configured globally and the VRF definitions can help to distinguish them.</td>
</tr>
<tr>
<td><strong>Step 5</strong> Use the commit or end command.</td>
<td>commit—Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>end—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>
### 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 vrf vrf-name**
7. Repeat Step 2, on page 31 through Step 5, on page 32 for each external server to be configured.
8. Use the **commit** or **end** command.
9. **show tacacs**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> tacacs-server host host-name port port-number</td>
<td>Specifies a TACACS+ host server and optionally specifies a server port number.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 port 51 RP/0/RSP0/CPU0:router(config-tacacs-host)#</td>
<td>• This option overrides the default, port 49. Valid port numbers range from 1 to 65535.</td>
</tr>
<tr>
<td><strong>Step 3</strong> tacacs-server host host-name timeout seconds</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-tacacs-host)# tacacs-server</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**  
```bash  
tacacs-server host host-name key [0 | 7] auth-key  
```

**Example:**  
```bash  
RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 key 0 a_secret  
```

### Purpose

- 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.

- 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.
  - (Optional) Entering 0 indicates that an unencrypted (clear-text) key follows.
  - (Optional) Entering 7 indicates that an encrypted key follows.
  - The `auth-key` argument specifies the encrypted or unencrypted key to be shared between the AAA server and the TACACS+ server.

**Step 5**  
```bash  
tacacs-server host host-name single-connection  
```

**Example:**  
```bash  
RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 single-connection  
```

### Purpose

- 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.

**Step 6**  
```bash  
tacacs source-interface type instance vrf vrf-name  
```

**Example:**  
```bash  
RP/0/RSP0/CPU0:router(config)# tacacs source-interface GigabitEthernet 0/4/0/0 vrf abc  
```

### Purpose

- (Optional) Specifies the source IP address of a selected interface for all outgoing TACACS+ packets.
  - 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 31 through Step 5, on page 32 for each external server to be configured.

**Step 8**  
Use the `commit` or `end` command.

- `commit`—Saves the configuration changes and remains within the configuration session.
### Purpose

Command or Action | Purpose
--- | ---
end—Prompts user to take one of these actions:
- **Yes**—Saves configuration changes and exits the configuration session.
- **No**—Exits the configuration session without committing the configuration changes.
- **Cancel**—Remains in the configuration mode, without committing the configuration changes.

### Step 9

**show tacacs**

**Example:**

```
RP/0/RSP0/CPU0:router# show tacacs
```

(Optimal) Displays information about the TACACS+ servers that are configured in the system.

### What to Do Next

After configuring TACACS+ servers, configure TACACS+ server groups. (See the Configuring TACACS+ Server Groups, on page 35 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 8 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} [auth-port port-number] [acct-port port-number]
4. Repeat Step 4, on page 34 for every external server to be added to the server group named in Step 3, on page 34.
5. deadtime minutes
6. Use the commit or end command.
7. show radius server-groups [group-name [detail]]
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> &lt;br&gt;RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> aaa group server radius <strong>group-name</strong></td>
<td>Groups different server hosts into distinct lists and enters the server group configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> &lt;br&gt;RP/0/RSP0/CPU0:router(config)# aaa group server radius radgroup1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> server {hostname</td>
<td>ip-address} [auth-port port-number] [acct-port port-number]</td>
</tr>
<tr>
<td><strong>Example:</strong> &lt;br&gt;RP/0/RSP0/CPU0:router(config-sg-radius)# server 192.168.20.0</td>
<td>• After the server group is configured, it can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).</td>
</tr>
<tr>
<td><strong>Step 4</strong> Repeat <strong>Step 4</strong>, on page 34 for every external server to be added to the server group named in <strong>Step 3</strong>, on page 34.</td>
<td>—</td>
</tr>
<tr>
<td><strong>Step 5</strong> deadline <strong>minutes</strong></td>
<td>Configures the deadtime value at the RADIUS server group level.</td>
</tr>
<tr>
<td><strong>Example:</strong> &lt;br&gt;RP/0/RSP0/CPU0:router(config-sg-radius)# deadline 1</td>
<td>• The <strong>minutes</strong> 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.</td>
</tr>
<tr>
<td><strong>Note</strong> You can configure the group-level deadtime after the group is created.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>Note</strong> You can configure the group-level deadtime after the group is created.</td>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Cancel</strong>—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>
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 8 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}`
4. Repeat Step 3, on page 36 for every external server to be added to the server group named in Step 2, on page 36.
5. `server-private {hostname | ip-address} [port port-number] [timeout seconds] [key string]`
6. `vrf vrf-name`
7. Use the `commit` or `end` command.
8. `show tacacs server-groups`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>aaa group server tacacs+ group-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>aaa group server tacacs+ tacgroup1</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>**server {hostname</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>server 192.168.100.0</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat <strong>Step 3, on page 36</strong> for every external server to be added to the server group named in <strong>Step 2, on page 36.</strong></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**server-private {hostname</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>server-private 10.1.1.1 key a_secret</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>vrf vrf-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>vrf abc</code></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td></td>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring AAA Services

#### 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 and aux 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 and aux 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**

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

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 | ppp} {default | list-name | remote} 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> aaa authentication {login</td>
<td>ppp} {default</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config)# aaa authentication login default group tacacs+</code></td>
<td></td>
</tr>
</tbody>
</table>

- 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 **list-name** character string identifies the authentication method list.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Entering the <strong>remote</strong> keyword causes the listed authentication methods that follow this keyword to be the default list of methods for administrative authentication on a remote non-owner SDR.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The remote keyword is available only on the admin plane.</td>
</tr>
<tr>
<td>• Entering a <strong>method-list</strong> 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:</td>
<td></td>
</tr>
<tr>
<td>◦ <strong>group tacacs</strong>—Use a server group or TACACS+ servers for authentication</td>
<td></td>
</tr>
<tr>
<td>◦ <strong>group radius</strong>—Use a server group or RADIUS servers for authentication</td>
<td></td>
</tr>
<tr>
<td>◦ <strong>group named-group</strong>—Use a named subset of TACACS+ or RADIUS servers for authentication</td>
<td></td>
</tr>
<tr>
<td>◦ <strong>local</strong>—Use a local username or password database for authentication</td>
<td></td>
</tr>
<tr>
<td>◦ <strong>line</strong>—Use line password or user group for authentication</td>
<td></td>
</tr>
<tr>
<td>• The example specifies the <strong>default</strong> method list to be used for authentication.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</td>
<td></td>
</tr>
<tr>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
<td></td>
</tr>
<tr>
<td>• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
<td></td>
</tr>
<tr>
<td>• <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Cancel</strong>—Remains in the configuration mode, without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat Step 1 through Step 3 for every authentication method list to be configured.</td>
</tr>
</tbody>
</table>

**What to Do Next**
After configuring authentication method lists, configure authorization method lists. (See the Configuring Authorization Method Lists, on page 39 section).

**Configuring Authorization Method Lists**
This task configures method lists for authorization.
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 Cisco IOS XR software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS XR 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.

The Cisco IOS XR 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. Cisco IOS XR software supports four types of AAA authorization:

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

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

- **EXEC authorization**—Applies authorization for starting EXEC session.

  *Note* The `exec` keyword is no longer used to authorize the fault manager service. The `eventmanager` keyword (fault manager) is used to authorize the fault manager service. The `exec` keyword is used for EXEC authorization.

- **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 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RPI/RSP0/CPU0:router#&lt;br&gt;configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>`aaa authorization {commands</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>`RP/0/RSP0/CPU0:router(config)#</td>
<td>• The <strong>eventmanager</strong> keyword applies an authorization method for authorizing an event manager (fault manager).</td>
</tr>
<tr>
<td>`aaa authorization commands</td>
<td>• The <strong>exec</strong> keyword configures authorization for an interactive (EXEC) session.</td>
</tr>
<tr>
<td>`listname1 group tacacs+</td>
<td>• The <strong>network</strong> keyword configures authorization for network services like PPP or IKE.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>default</strong> keyword causes the listed authorization methods that follow this keyword to be the default list of methods for authorization.</td>
</tr>
<tr>
<td></td>
<td>• A <strong>list-name</strong> character string identifies the authorization method list. The method list itself follows the method list name. Method list types are entered in the preferred sequence. The listed method list types can be any one of the following:</td>
</tr>
<tr>
<td></td>
<td>• <strong>none</strong>—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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>local</strong>—Uses local database for authorization.</td>
</tr>
<tr>
<td>Step 3</td>
<td></td>
</tr>
<tr>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>

**What to Do Next**

After configuring authorization method lists, configure accounting method lists. (See the Configuring Accounting Method Lists, on page 43 section.)
Configuring Accounting Method Lists

This task configures method lists for accounting.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

<table>
<thead>
<tr>
<th><strong>Step 2</strong> Do one of the following:</th>
<th>Creates a series of accounting methods, or a method list.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• aaa accounting {commands</td>
<td>• The <strong>commands</strong> keyword enables accounting for EXEC shell commands.</td>
</tr>
<tr>
<td></td>
<td>• exec</td>
</tr>
<tr>
<td></td>
<td>network</td>
</tr>
<tr>
<td></td>
<td>{default</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# aaa accounting commands default stop-only group tacacs+
```

<table>
<thead>
<tr>
<th><strong>Step 3</strong> Use the <strong>commit</strong> or <strong>end</strong> command.</th>
<th><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
<td><strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td>• <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
<td></td>
</tr>
</tbody>
</table>
What to Do Next

After configuring method lists, apply those method lists. (See the Applying Method Lists for Applications, on page 46 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `aaa accounting update {newinfo</td>
<td>periodic minutes}`</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# aaa accounting update periodic 30</td>
<td>• If the <code>newinfo</code> keyword is used, interim accounting records are sent to the accounting server every time 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.</td>
</tr>
<tr>
<td></td>
<td>• When used with the <code>periodic</code> keyword, interim accounting records are sent periodically as defined by the argument number. The interim accounting</td>
</tr>
</tbody>
</table>
Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>interim accounting record</td>
<td>Contains all the accounting information recorded for that user up to the time the interim accounting record is sent.</td>
</tr>
<tr>
<td>Caution</td>
<td>The periodic keyword causes heavy congestion when many users are logged in to the network.</td>
</tr>
</tbody>
</table>

**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:

- **Yes**—Saves configuration changes and exits the configuration session.
- **No**—Exits the configuration session without committing the configuration changes.
- **Cancel**—Remains in the configuration mode, 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, auxiliary, 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 {aux | console | default | template template-name}**
3. **authorization {commands | exec} {default | list-name}**
4. Use the **commit** or **end** command.
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> line {aux</td>
<td>console</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# line console</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> authorization {commands</td>
<td>exec} {default</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-line)# authorization commands listname5</td>
<td>• The <strong>commands</strong> keyword enables authorization on the selected lines for all commands.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>exec</strong> keyword enables authorization for an interactive (EXEC) session.</td>
</tr>
<tr>
<td></td>
<td>• Enter the <strong>default</strong> keyword to apply the name of the default method list, as defined with the <strong>aaa authorization</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• 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 <strong>aaa authorization</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• The example enables command authorization using the method list named listname5.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</td>
<td></td>
</tr>
<tr>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Cancel</strong>—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>

### 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 48 section.)
Enabling Accounting Services

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

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>line { aux</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# line console</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters line template configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>accounting {commands</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-line)# accounting commands listname7</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enables AAA accounting for a specific line or group of lines.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>commands</strong> keyword enables accounting on the selected lines for all EXEC shell commands.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>exec</strong> keyword enables accounting for an interactive (EXEC) session.</td>
</tr>
<tr>
<td></td>
<td>• Enter the <strong>default</strong> keyword to apply the name of the default method list, as defined with the <strong>aaa accounting</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• 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 <strong>aaa accounting</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• The example enables command accounting using the method list named listname7.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td><strong>end</strong>—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
</tr>
</tbody>
</table>
What to Do Next

After applying accounting method lists by enabling AAA accounting services, configure login parameters. (See the Configuring Login Parameters, on page 49 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>line template template-name</code></td>
<td>Specifies a line to configure and enters line template configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# line template alpha</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>timeout login response seconds</code></td>
<td>Sets the interval that the server waits for reply to a login.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-line)# timeout login response 20</td>
</tr>
</tbody>
</table>

- The `seconds` 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.
<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
|        | 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:  
• **Yes**—Saves configuration changes and exits the configuration session.  
• **No**—Exits the configuration session without committing the configuration changes.  
• **Cancel**—Remains in the configuration mode, 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.

```plaintext
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`.

```plaintext
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`.

```plaintext
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 10.1.1.1 port 1 key abc
tacacs-server host 10.2.2.2 port 2 key def
tacacs-server host 10.3.3.3 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 10.3.3.3
exit

Additional References

The following sections provide references related to configuring AAA services.
## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA services commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td><em>Authentication, Authorization, and Accounting Commands on the Cisco ASR 9000 Series Router in the</em></td>
</tr>
</tbody>
</table>

## Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
CHAPTER 2

Implementing Certification Authority Interoperability

Certification authority (CA) interoperability is provided in support of the IP Security (IPSec), Secure Socket Layer (SSL), and Secure Shell (SSH) protocols. This module describes how to implement CA interoperability.

CA interoperability permits Cisco ASR 9000 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

IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).

For a complete description of the public key infrastructure (PKI) commands used in this chapter, refer to the Public Key Infrastructure Commands module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference. To locate documentation for other commands that appear in this module, use the command reference master index, or search online.

Feature History for Implementing Certification Authority Interoperability

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 5.2</td>
<td>A section was added on trust pool management</td>
</tr>
<tr>
<td>Release 5.2</td>
<td>A section was added on trust pool management</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing Certification Authority, page 54
- Restrictions for Implementing Certification Authority, page 54
- Information About Implementing Certification Authority, page 54
- How to Implement CA Interoperability, page 57
- Configuration Examples for Implementing Certification Authority Interoperability, page 65
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 Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide

- 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.

  Note: IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).

- **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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Example: <code>configure</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Configures the hostname of the router.</td>
</tr>
<tr>
<td><code>hostname name</code></td>
<td>Example: <code>hostname myhost</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config)# hostname myhost</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures the IP domain name of the router.</td>
</tr>
<tr>
<td><code>domain name domain-name</code></td>
<td>Example: <code>domain name mydomain.com</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config)# domain name mydomain.com</code></td>
</tr>
</tbody>
</table>
| **Step 4**                      |                            | commit—Saves the configuration changes and remains within the configuration session. end—Prompts user to take one of these actions:
| Use the `commit` or `end` command. |                           |
| **Example:**                    |                            | • Yes—Saves configuration changes and exits the configuration session.   |
| **Example:**                    |                            | • No—Exits the configuration session without committing the configuration changes. |
|                                |                            | • Cancel—Remains in the configuration mode, 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>crypto key generate rsa [usage keys</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# crypto key generate rsa general-keys</td>
</tr>
<tr>
<td></td>
<td>Generates RSA key pairs.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>usage keys</strong> keyword to specify special usage keys; use the <strong>general-keys</strong> keyword to specify general-purpose RSA keys.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>keypair-label</strong> argument is the RSA key pair label that names the RSA key pairs.</td>
</tr>
</tbody>
</table>

| Step 2 | crypto key zeroize rsa [keypair-label] |
| **Example:** | RP/0/RSP0/CPU0:router# crypto key zeroize rsa key1 |
| | (Optional) Deletes all RSAs from the router. |
| | • 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. |
| | • To remove a specific RSA key pair, use the **keypair-label** argument. |

| Step 3 | show crypto key mypubkey rsa |
| **Example:** | RP/0/RSP0/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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Generates RSA key pairs.</td>
</tr>
<tr>
<td>crypto key import authentication rsa [usage keys</td>
<td>• Use the usage keys keyword to specify special usage keys; use the general-keys keyword to specify general-purpose RSA keys.</td>
</tr>
<tr>
<td>general-keys] [keypair-label]</td>
<td>• The keypair-label argument is the RSA key pair label that names the RSA key pairs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# crypto key import authentication rsa general-keys</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>(Optional) Displays the RSA public keys for your router.</td>
</tr>
<tr>
<td>show crypto key mypubkey rsa</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# show crypto key mypubkey rsa</td>
<td></td>
</tr>
</tbody>
</table>

### 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. rsakeypair keypair-label
8. Use the commit or end command.

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Declares a CA.</td>
</tr>
<tr>
<td>crypto ca trustpoint ca-name</td>
<td></td>
</tr>
</tbody>
</table>

---

Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 4.1

[50x870]
### Implementing Certification Authority Interoperability

#### Declaring a Certification Authority and Configuring a Trusted Point

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint myca</td>
<td>• Configures a trusted point with a selected name so that your router can verify certificates issued to peers. • Enlists trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> enrollment url CA-URL</td>
<td>Specifies the URL of the CA.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-trustp)# enrollment url <a href="http://ca.domain.com/certsrv/mscep/mscep.dll">http://ca.domain.com/certsrv/mscep/mscep.dll</a></td>
<td>• The URL should include any nonstandard cgi-bin script location.</td>
</tr>
<tr>
<td><strong>Step 4</strong> query url LDAP-URL</td>
<td>(Optional) Specifies the location of the LDAP server if your CA system supports the LDAP protocol.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> enrollment retry period minutes</td>
<td>(Optional) Specifies a retry period.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-trustp)# enrollment retry period 2</td>
<td>• 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.</td>
</tr>
<tr>
<td><strong>Step 6</strong> enrollment retry count number</td>
<td>(Optional) Specifies how many times the router continues to send unsuccessful certificate requests before giving up.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-trustp)# enrollment retry count 10</td>
<td>• The range is from 1 to 100.</td>
</tr>
<tr>
<td><strong>Step 7</strong> rsakeypair keypair-label</td>
<td>(Optional) Specifies a named RSA key pair generated using the <strong>crypto key generate rsa</strong> command for this trustpoint.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-trustp)# rsakeypair mykey</td>
<td>• Not setting this key pair means that the trustpoint uses the default RSA key in the current configuration.</td>
</tr>
<tr>
<td><strong>Step 8</strong> Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session. <strong>end</strong>—Prompts user to take one of these actions: • <strong>Yes</strong>—Saves configuration changes and exits the configuration session. • <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
</tr>
</tbody>
</table>
Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Cancel</td>
<td>Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>crypto ca authenticate ca-name</code></td>
<td>Authenticates the CA to your router by obtaining a CA certificate, which contains the public key for the CA.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# crypto ca authenticate myca</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>show crypto ca certificates</code></td>
<td>(Optional) Displays information about the CA certificate.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# show crypto ca certificates</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>crypto ca enroll ca-name</td>
<td>Requests certificates for all of your RSA key pairs.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# crypto ca enroll myca</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• This command causes your router to request as many certificates as</td>
</tr>
<tr>
<td></td>
<td>there are RSA key pairs, so you need only perform this command once,</td>
</tr>
<tr>
<td></td>
<td>even if you have special usage RSA key pairs.</td>
</tr>
<tr>
<td></td>
<td>• This command requires you to create a challenge password that is not</td>
</tr>
<tr>
<td></td>
<td>saved with the configuration. This password is required if your</td>
</tr>
<tr>
<td></td>
<td>certificate needs to be revoked, so you must remember this password.</td>
</tr>
<tr>
<td></td>
<td>• A certificate may be issued immediately or the router sends a</td>
</tr>
<tr>
<td></td>
<td>certificate request every minute until the enrollment retry period is</td>
</tr>
<tr>
<td></td>
<td>reached and a timeout occurs. If a timeout occurs, contact your</td>
</tr>
<tr>
<td></td>
<td>system administrator to get your request approved, and then enter this</td>
</tr>
<tr>
<td></td>
<td>command again.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>(Optional) Displays information about the CA certificate.</td>
</tr>
<tr>
<td>show crypto ca certificates</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# show crypto ca certificates</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>crypto ca trustpoint ca-name</strong>&lt;br&gt;Declares the CA that your router should use and enters trustpoint configuration mode.&lt;br&gt;• Use the <code>ca-name</code> argument to specify the name of the CA.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint myca&lt;br&gt;RP/0/RSP0/CPU0:router(config-trustp)#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>enrollment terminal</strong>&lt;br&gt;Specifies manual cut-and-paste certificate enrollment.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-trustp)# enrollment terminal</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.&lt;br&gt;<strong>commit</strong>—Saves the configuration changes and remains within the configuration session.&lt;br&gt;<strong>end</strong>—Prompts user to take one of these actions:&lt;br&gt;• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.&lt;br&gt;• <strong>No</strong>—Exits the configuration session without committing the configuration changes.&lt;br&gt;• <strong>Cancel</strong>—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>crypto ca authenticate ca-name</strong>&lt;br&gt;Authenticates the CA by obtaining the certificate of the CA.&lt;br&gt;• Use the <code>ca-name</code> argument to specify the name of the CA. Use the same name that you entered in <strong>Step 2</strong>, on page 64.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# crypto ca authenticate myca</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>crypto ca enroll ca-name</strong>&lt;br&gt;Obtains the certificates for your router from the CA.&lt;br&gt;• Use the <code>ca-name</code> argument to specify the name of the CA. Use the same name that you entered in <strong>Step 2</strong>.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# crypto ca enroll myca</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>crypto ca import ca-name certificate</strong>&lt;br&gt;Imports a certificate manually at the terminal.&lt;br&gt;• Use the <code>ca-name</code> argument to specify the name of the CA. Use the same name that you entered in <strong>Step 2</strong>.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# crypto ca import myca certificate</td>
</tr>
</tbody>
</table>
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.)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>crypto ca import</code></td>
<td>You must enter the <code>crypto ca import</code> 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.)</td>
</tr>
</tbody>
</table>

**Step 8**

 display crypto ca certificates

**Example:**

```plaintext
RP/0/RSP0/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.

```plaintext
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 B7DEB8F8 1C54AF7F 293F3004
```
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
```

Password:
Re-enter Password:
  Fingerprint: 17D8B38D ED2BDF2E DF8ADBF7 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. IKE configuration is described in the Implementing Internet Key Exchange Security Protocol on the Cisco ASR 9000 Series Router module, IPSec in the Implementing IPsec Network Security on the Cisco ASR 9000 Series Router module, and SSL in the Implementing Secure Socket Layer on the Cisco ASR 9000 Series Router module. These modules are located in Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide (this publication).

Additional References

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

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Public Key Infrastructure Commands on the Cisco ASR 9000 Series Router module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>
### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Configuring Keychain Management, page 69
- Restrictions for Implementing Keychain Management, page 69
- Information About Implementing Keychain Management, page 70
- How to Implement Keychain Management, page 71
- Configuration Examples for Implementing Keychain Management, page 80
- Additional References, page 81

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

- Resource Reservation Protocol (RSVP) uses keychain for authentication. For more information about RSVP, see the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide.

- IP Service Level Agreements (IP SLAs) use a keychain for MD5 authentication for the IP SLA control message. For more information about IP SLAs, see the Cisco ASR 9000 Series Aggregation Services Router System Monitoring Configuration Guide and the key-chain command in the Cisco ASR 9000 Series Aggregation Services Router System Monitoring Command Reference.

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

Lifetime of a 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> key chain key-chain-name</td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>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 global configuration mode attributes or keychain-key configuration mode attributes (for example, lifetime or key string).</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys RP/0/RSP0/CPU0:router(config-isis-keys)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> Use the commit or end command.</td>
<td>commit—Saves the configuration changes and remains within the configuration session. end—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td><strong>Step 4</strong> show key chain key-chain-name</td>
<td>(Optional) Displays the name of the keychain.</td>
</tr>
</tbody>
</table>
### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> key chain <em>key-chain-name</em></td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> accept-tolerance <em>value [infinite]</em></td>
<td>Configures a tolerance value to accept keys for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
| RP/0/RSP0/CPU0:router(config-isis-keys)# accept-tolerance infinite | • Use the value 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. |

**What to Do Next**

After completing keychain configuration, see the Configuring a Tolerance Specification to Accept Keys, on page 72 section.
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 key chain</td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td>key-chain-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>key</strong> key-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)# key 8</td>
</tr>
<tr>
<td></td>
<td>Creates a key for the keychain. The key ID number is translated from decimal to hexadecimal to create the command mode subprompt.</td>
</tr>
<tr>
<td></td>
<td>• Use the <em>key-id</em> argument as a 48-bit integer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Command or Action</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td><strong>commit</strong></td>
<td>Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td>Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Cancel</strong>—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>

**What to Do Next**

After configuring a key identifier for the keychain, see the Configuring the Text for the Key String, on page 74 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>key chain <em>key-chain-name</em></td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>key <em>key-id</em></td>
<td>Creates a key for the keychain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)# key &amp; RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>key-string [clear</td>
<td>password] <em>key-string-text</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# key-string password 8</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td></td>
</tr>
</tbody>
</table>

### commit
-Saves the configuration changes and remains within the configuration session.

### end
-Prompts user to take one of these actions:
  - **Yes**—Saves configuration changes and exits the configuration session.

---

**Implementing Keychain Management**

**Configuring the Text for the Key String**
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> key chain <em>key-chain-name</em></td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> key <em>key-id</em></td>
<td>Creates a key for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)# key</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**accept-lifetime ** <code>start-time</code> [duration <code>duration-value</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# accept-lifetime 1:00:00 october 24 2005 infinite</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>key chain key-chain-name</td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
</tr>
<tr>
<td>Step 3</td>
<td>key key-id</td>
<td>Creates a key for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)# key 8 RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#</td>
</tr>
</tbody>
</table>
| Step 4 | send-lifetime start-time [duration duration-value | (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:  
  • duration keyword (seconds)  
  • infinite keyword  
  • end-time argument |
| Example: | infinite | end-time] | |
| | | RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# send-lifetime 1:00:00 october 24 2005 infinite |
| 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:  
  • Yes—Saves configuration changes and exits the configuration session.  
  • No—Exits the configuration session without committing the configuration changes.  
  • Cancel—Remains in the configuration mode, without committing the configuration changes. |
Configuring the Cryptographic Algorithm

This task allows the keychain 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> key chain key-chain-name</td>
<td>Creates a name for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# key chain isis-keys</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> key key-id</td>
<td>Creates a key for the keychain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-keys)# key</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cryptographic-algorithm [HMAC-MD5</td>
<td>HMAC-SHA1-12</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# cryptographic-algorithm MD5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Border Gateway Protocol (BGP) supports only HMAC-MD5 and HMAC-SHA1-12.</td>
<td></td>
</tr>
<tr>
<td>• Intermediate System-to-Intermediate System (IS-IS) supports only HMAC-MD5.</td>
<td></td>
</tr>
<tr>
<td>• Open Shortest Path First (OSPF) supports only MD5 and HMAC-MD5.</td>
<td></td>
</tr>
</tbody>
</table>

### 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:
  - **Yes**—Saves configuration changes and exits the configuration session.
  - **No**—Exits the configuration session without committing the configuration changes.
  - **Cancel**—Remains in the configuration mode, 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 mykey91abcd
  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
```
### Additional References

The following sections provide references related to implementing keychain management.

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keychain management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Keychain Management Commands in the Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>able technical content, including links to products, technologies, solutions,</td>
<td></td>
</tr>
<tr>
<td>technical tips, and tools. Registered Cisco.com users can log in from this</td>
<td></td>
</tr>
<tr>
<td>page to access even more content.</td>
<td></td>
</tr>
</tbody>
</table>
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.

Feature History for Implementing Lawful Intercept

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.1.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing Lawful Intercept, page 83
- Restrictions for Implementing Lawful Intercept, page 85
- Information About Lawful Intercept Implementation, page 85
- How to Configure SNMPv3 Access for Lawful Intercept on the Cisco ASR 9000 Series Router, page 89
- Configuration Example for Inband Management Plane Feature Enablement, page 92
- Additional References, page 93

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:
Cisco ASR 9000 Series Aggregation Services Router will be used as content Intercept Access Point (IAP) router in lawful interception operation.

Provisioned router—The router must be already provisioned. For more information, see Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide.

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 Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference. 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 Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide.

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

Lawful intercept does not provide support for these features on Cisco ASR 9000 Series Router:

- IPv6 multicast tapping
- IPv4 multicast tapping
- Per tap drop counter
- IPv6 MD encapsulation
- Per interface tapping
- Replicating a single tap to multiple MDs
- Tapping of tag packets
- Tapping L2 flows
- RTP encapsulation
- Encryption and integrity checking of replication device

Note

Per tap drop counter support is available only for ASR9000-SIP-700 line card, and not for ethernet line cards.

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

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 86.)

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*

![Diagram of Lawful Intercept Topology](image)

Scale or Performance Improvement

New enhancements introduced on the Cisco ASR 9000 Series Router in terms of scalability and performance for lawful intercept are:

- IPv4 lawful intercept tap limit is 1000 taps per IPv4.
- IPv6 lawful intercept tap limit is 1000 taps per IPv6.
- Interception rate is:
  - 50 Mbps per network processor (NP) for ASR9000-SIP-700 line card.
  - 100 Mbps for Gigabit Ethernet line cards.
- Support up to 512 MDs.
How to Configure SPNPv3 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 ASR 9000 Series Router.

• 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/RSP0/CP00:router# configure
RP/0/RSP0/CP00:router(config)# lawful-intercept disable
```

Note: 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 96.

Related Tasks

• Configuring a Device for Management Plane Protection for an Inband Interface, on page 98

Related Examples

• Configuring the Inband Management Plane Protection Feature: Example, on page 92
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>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.</td>
</tr>
<tr>
<td>snmp-server view view-name ciscoTap2MIB included</td>
<td>RP/0//CPU0:router(config)# snmp-server view TapName ciscoTap2MIB included</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>RP/0//CPU0:router(config)# snmp-server view TapName ciscoUserConnectionTapMIB included</td>
</tr>
<tr>
<td>snmp-server view view-name ciscoUserConnectionTapMIB included</td>
<td>RP/0//CPU0:router(config)# snmp-server view TapName ciscoUserConnectionTapMIB included</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>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.</td>
</tr>
<tr>
<td>snmp-server group group-name v3auth read view-name write view-name notify view-name</td>
<td>RP/0//CPU0:router(config)# snmp-server</td>
</tr>
</tbody>
</table>
### Purpose

Command or Action | Purpose
--- | ---
`group TapGroup v3 auth read TapView write TapView notify TapView` | Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><code>snmp-server host ip-address traps version 3 auth username udp-port port-number</code></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/CPU0:router(config)# snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555</code></td>
<td>• The <code>mduser-id</code> and <code>mdpassword</code> must match that configured on MD. Alternatively, these values must match those in use on the router.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Passwords must be eight characters or longer to comply with SNMPv3 security minimums.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Minimum Lawful Intercept security level is auth; The <code>noauth</code> option will not work, as it indicates noAuthnoPriv security level. The Lawful Intercept security level must also match that of the MD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Choices other than MD5 are available on the router, but the MD values must match. Most MDs default to or support only MD5.</td>
</tr>
<tr>
<td>6</td>
<td><code>snmp-server user mduser-id groupname v3 auth md5 md-password</code></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/CPU0:router(config)# snmp-server user TapGroup v3 auth md5 mdpassword</code></td>
<td>commit—Saves the configuration changes and remains within the configuration session. end—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
<tr>
<td>7</td>
<td>Use the <code>commit</code> or <code>end</code> command.</td>
<td>Displays information about each SNMP username in the SNMP user table.</td>
</tr>
<tr>
<td>8</td>
<td><code>show snmp users</code></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/CPU0:router# show snmp users</code></td>
<td></td>
</tr>
</tbody>
</table>
### 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-ctrl)# 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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lawful Intercept commands</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
<tr>
<td>Implementing SNMP</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide</td>
</tr>
<tr>
<td>SNMP Server commands</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>A modular, open architecture designed for simple implementation that easily interacts with third-party equipment to meet service provider lawful intercept requirements.</td>
<td>See RFC-3924 under RFCs, on page 93.</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CISCO-TAP2-MIB, version 2</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
<tr>
<td>• CISCO-IP-TAP-MIB</td>
<td></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC-3924</td>
<td>Cisco Architecture for Lawful Intercept in IP Networks</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
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.

This module describes how to implement management plane protection on Cisco ASR 9000 Series Routers.

For information on MPP commands, see the Management Plane Protection Commands module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.

Feature History for Implementing Management Plane Protection

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing Management Plane Protection, page 95
- Restrictions for Implementing Management Plane Protection, page 96
- Information About Implementing Management Plane Protection, page 96
- How to Configure a Device for Management Plane Protection, page 98
- Configuration Examples for Implementing Management Plane Protection, page 104
- Additional References, page 105

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 Multiprotocol Label Switching (MPLS) Virtual private Network (VPN), the VPN Routing and Forwarding (VRF) is globally configured for the ingress interface of the Provider Edge (PE) router. MPP applies the VRF on the interface filters. When an incoming packet from the core interface has a different VRF, then MPP does not work in MPLS VPN.

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 in-band, 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
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>control-plane</code></td>
<td>Enters control plane configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# control-plane RP/0/RSP0/CPU0:router(config-ctrl)#</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>management-plane</code></td>
<td>Configures management plane protection to allow and disallow protocols and enters management plane protection configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ctrl)# management-plane RP/0/RSP0/CPU0:router(config-mpp)#</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>inband</code></td>
<td>Configures an inband interface and enters management plane protection inband configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpp)# inband RP/0/RSP0/CPU0:router(config-mpp-inband)#</td>
</tr>
<tr>
<td>Step 5</td>
<td>`interface {type instance</td>
<td>all}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>Use the <code>all</code> keyword to configure all interfaces.</td>
</tr>
</tbody>
</table>
### Configuring a Device for Management Plane Protection for an Inband Interface

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpp-inband)# interface GigabitEthernet 0/6/0/1 RP/0/RSP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)#</td>
<td>Configures an interface as an inband interface for a specified protocol or all protocols.</td>
</tr>
<tr>
<td><strong>Step 6</strong> allow {protocol</td>
<td>all} [peer]</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)# allow Telnet peer RP/0/RSP0/CPU0:router(config-telnet-peer)#</td>
<td>HTTP or HTTPS, SNMP (also versions), Secure Shell (v1 and v2), TFTP, Telnet</td>
</tr>
<tr>
<td><strong>Step 7</strong> address ipv4 {peer-ip-address</td>
<td>peer ip-address/length}</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-telnet-peer)# address ipv4 10.1.0.0/16</td>
<td>Use the peer ip-address/length argument to configure the prefix of the peer IPv4 address.</td>
</tr>
<tr>
<td><strong>Step 8</strong> Use the commit or end command.</td>
<td>commit—Saves the configuration changes and remains within the configuration session. end—Prompts user to take one of these actions: Yes—Saves configuration changes and exits the configuration session. No—Exits the configuration session without committing the configuration changes. Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
<tr>
<td><strong>Step 9</strong> show mgmt-plane [inband</td>
<td>out-of-band] [interface {type instance}]</td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2**        | Enters control plane configuration mode. |
| `control-plane`   |         |
| **Example:**      |         |
| RP/0/RSP0/CPU0:router(config)# control-plane | RP/0/RSP0/CPU0:router(config-ctrl)# |

| **Step 3**        | Configures management plane protection to allow and disallow protocols and enters management plane protection configuration mode. |
| `management-plane`|         |
| **Example:**      |         |
| RP/0/RSP0/CPU0:router(config-ctrl)# management-plane | RP/0/RSP0/CPU0:router(config-mpp)# |

| **Step 4**        | Configures out-of-band interfaces or protocols and enters management plane protection out-of-band configuration mode. |
| `out-of-band`     |         |
| **Example:**      |         |
| RP/0/RSP0/CPU0:router(config-mpp)# out-of-band | RP/0/RSP0/CPU0:router(config-mpp-outband)# |

| **Step 5**        | Configures a Virtual Private Network (VPN) routing and forwarding (VRF) reference of an out-of-band interface. |
| `vrf vrf-name`    | • Use the `vrf-name` argument to assign a name to a VRF. |
| **Example:**      |         |
| RP/0/RSP0/CPU0:router(config-mpp-outband)# vrf target | |

| **Step 6**        | 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. |
| `interface {type instance | all}` | • Use the `all` keyword to configure all interfaces. |
| **Example:**      |         |
| RP/0/RSP0/CPU0:router(config-mpp-outband)# interface GigabitEthernet 0/6/0/2 | RP/0/RSP0/CPU0:router(config-mpp-outband-Gi0_6_0_2)# |
### Configuring a Device for Management Plane Protection for an Out-of-band Interface

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td>Configure an interface as an out-of-band interface for a specified protocol or all protocols.</td>
</tr>
<tr>
<td>`allow [protocol</td>
<td>all] [peer]`</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-mpp-outband-Gi0_6_0_2)# allow TFTP peer RP/0/RSP0/CPU0:router(config-tftp-peer)#</td>
<td>Use the <code>protocol</code> argument to allow management protocols on the designated management interface.</td>
</tr>
<tr>
<td></td>
<td>- HTTP or HTTPS</td>
</tr>
<tr>
<td></td>
<td>- SNMP (also versions)</td>
</tr>
<tr>
<td></td>
<td>- Secure Shell (v1 and v2)</td>
</tr>
<tr>
<td></td>
<td>- TFTP</td>
</tr>
<tr>
<td></td>
<td>- Telnet</td>
</tr>
<tr>
<td></td>
<td>Use the <code>all</code> keyword to configure the interface to allow all the management traffic that is specified in the list of protocols.</td>
</tr>
<tr>
<td></td>
<td>(Optional) Use the <code>peer</code> keyword to configure the peer address on the interface.</td>
</tr>
</tbody>
</table>

| Step 8 | Configure the peer IPv6 address in which management traffic is allowed on the interface. |
| `address ipv6 {peer-ip-address | peer ip-address/length}` | Configures the peer IPv6 address in which management traffic is allowed on the interface. |
| Example: RP/0/RSP0/CPU0:router(config-tftp-peer)# address ipv6 33::33 | Use the `peer-ip-address` argument to configure the peer IPv6 address in which management traffic is allowed on the interface. |
| | Use the `peer-ip-address/length` argument to configure the prefix of the peer IPv6 address. |

| Step 9 | 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: |
| | - **Yes** — Saves configuration changes and exits the configuration session. |
| | - **No** — Exits the configuration session without committing the configuration changes. |
| | - **Cancel** — Remains in the configuration mode, without committing the configuration changes. |

| Step 10 | Displays information about the management plane, such as type of interface and protocols enabled on the interface. |
| `show mgmt-plane [inband | out-of-band] [interface {type instance} | vrf]` | (Optional) Use the `inband` keyword to display the inband management interface configurations that are |
The interface that processes 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:

```plaintext
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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Management Plane Protection Commands on Cisco ASR 9000 Series Aggregation Services Router System Monitoring Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
**MIBs**

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

**RFCs**

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
CHAPTER 6

Configuring Software Authentication Manager

Software Authentication Manager (SAM) is a component of the Cisco ASR 9000 Series Router 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 Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.

For information on setting the system clock, see the clock set command in Clock Commands module in Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference.

Feature History for Configuring Software Authentication Manager

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Configuring Software Authentication Manager, page 107
- Information about Software Authentication Manager, page 107
- How to set up a Prompt Interval for the Software Authentication Manager, page 108

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 Packager 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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sam promptinterval</code> <code>time-interval</code> `{proceed</td>
<td>terminate}`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# sam prompt-interval 25 {proceed</td>
<td>terminate}</td>
</tr>
</tbody>
</table>
| **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:  
  • **Yes**—Saves configuration changes and exits the configuration session.  
  • **No**—Exits the configuration session without committing the configuration changes. |
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Cancel</strong></td>
<td>Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
</tbody>
</table>
How to set up a Prompt Interval for the Software Authentication Manager
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 `rexe` 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 Digital Signature Algorithm (DSA) keys. Cisco IOS XR software supports both SSHv1 and SSHv2.

This module describes how to implement Secure Shell on the the Cisco ASR 9000 Series Router.

For a complete description of the Secure Shell commands used in this module, see the `Secure Shell Commands` module in `Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference`.

### Feature History for Implementing Secure Shell

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
| Release 3.9.0 | Support was added for the following enhancements:  
  * RSA based authentication on the SSH server  
  * SFTP client in interactive mode  
  * SFTP server implementation |
| Release 5.1.1 | Support was added for SSH Multiplexing |

- Prerequisites to Implementing Secure Shell, page 112
- Restrictions for Implementing Secure Shell, page 112
- Information About Implementing Secure Shell, page 113
Prerequisites to 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 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 Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference publication and Configuring AAA Services on Cisco IOS XR Software module in the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide 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 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.

```bash
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.
  ```bash
  RP/0/5/CPU0:router#configure
  RP/0/5/CPU0:router(config)#ssh client dscp <value from 0 – 63>
  ```

- **Knownhost**—Enable the host pubkey check by local database.

- **Source-interface**—Source interface for SSH client sessions.
  ```bash
  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)
  MgntEth 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-IF 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
  ```

- **VRF**—Source interface VRF for SSH client sessions:
  ```bash
  RP/0/5/CPU0:router(config)#ssh client vrf ?
  ```

SSH also supports remote command execution as follows:

```bash
ssh <cr>
```
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 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 connection 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 2048 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 the Cisco ASR 9000 Series Router chapter in this guide.

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.

How to Implement Secure Shell

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

Configuring SSH

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]
   - ssh server v2
10. Use the commit or end command.
11. show ssh
12. show ssh session details

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>

Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 4.1
<table>
<thead>
<tr>
<th><strong>Step</strong></th>
<th><strong>Command or Action</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td><code>hostname hostname</code></td>
<td>Configures a hostname for your router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# hostname router1</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 3** | `domain name domain-name` | Defines a default domain name that the software uses to complete unqualified host names. |
| **Example:** | RP/0/RSP0/CPU0:router(config)# domain name cisco.com |

| **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:  
• Yes—Saves configuration changes and exits the configuration session.  
• No—Exits the configuration session without committing the configuration changes.  
• Cancel—Remains in the configuration mode, without committing the configuration changes. |

| **Step 5** | `crypto key generate rsa [usage keys | general-keys] [keypair-label]` | Generates an RSA key pair.  
  • To delete the RSA key pair, use the `crypto key zeroize rsa` command.  
  • This command is used for SSHv1 only. |
| **Example:** | RP/0/RSP0/CPU0:router# crypto key generate rsa general-keys |

| **Step 6** | `crypto key generate dsa` | Enables the SSH server for local and remote authentication on the router.  
  • The recommended minimum modulus size is 1024 bits.  
  • Generates a DSA key pair.  
  To delete the DSA key pair, use the `crypto key zeroize dsa` command.  
  • This command is used only for SSHv2. |
| **Example:** | RP/0/RSP0/CPU0:router# crypto key generate dsa |

| **Step 7** | `configure` | Enters global configuration mode. |
| **Example:** | RP/0/RSP0/CPU0:router# configure |

<p>| <strong>Step 8</strong> | <code>ssh timeout seconds</code> | (Optional) Configures the timeout value for user authentication to AAA. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:**<br>RP/0/RSP0/CPU0:router(config)# ssh timeout 60 | • If the user fails to authenticate itself to AAA within the configured time, the connection is aborted.  
• If no value is configured, the default value of 30 seconds is used. The range is from 5 to 120. |
| **Step 9**<br>Do one of the following:<br>• ssh server [vrf vrf-name]<br>• ssh server v2 | • (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. |
| **Example:**<br>RP/0/RSP0/CPU0:router(config)# ssh or<br>RP/0/RSP0/CPU0:router(config)# ssh server v2 | **Note**<br>The SSH server can be configured for multiple VRF usage.  
• (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**<br>Use the commit or end command. | commit—Saves the configuration changes and remains within the configuration session.<br>end—Prompts user to take one of these actions:<br>• Yes—Saves configuration changes and exits the configuration session.<br>• No—Exits the configuration session without committing the configuration changes.<br>• Cancel—Remains in the configuration mode, without committing the configuration changes. |
| **Step 11**<br>show ssh | (Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router. |
| **Example:**<br>RP/0/RSP0/CPU0:router# show ssh | **Step 12**<br>show ssh session details | (Optional) Displays a detailed report of the SSHv2 connections to and from the router. |
| **Example:**<br>RP/0/RSP0/CPU0:router# show ssh session details |
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 | hostname\} [ username user- id | cipher des | source-interface type instance\]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ssh client knownhost <em>device</em> : /filename</td>
<td>(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey</td>
<td>Note The complete path of the filename is required. The colon (:) and slash mark (/) are also required.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Use the commit or end command.</td>
<td>commit—Saves the configuration changes and remains within the configuration session. end—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• Yes—Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• No—Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Cancel—Remains in the configuration mode, without committing the configuration changes.</td>
</tr>
<tr>
<td><strong>Step 4</strong> ssh {ipv4-address</td>
<td>hostname} [ username user- id</td>
</tr>
<tr>
<td>Example:</td>
<td>• 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.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# ssh remotehost username user1234</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• The cipher des option can be used only with an SSHv1 client.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>• The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only.</td>
</tr>
<tr>
<td></td>
<td>• If the hostname argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.</td>
</tr>
</tbody>
</table>

- If you are using SSHv1 and your SSH connection is being rejected, you have not successfully generated an RSA key pair for your router. Make sure that you have specified a hostname and domain. Then use the `crypto key generate rsa` command to generate an RSA key pair and enable the SSH server.

- If you are using SSHv2 and your SSH connection is being rejected, you have not successfully generated a DSA key pair for your router. Make sure that you have specified a hostname and domain. Then use the `crypto key generate dsa` command to generate a DSA key pair and 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.

Make sure that you have specified a hostname and domain. Then use the `crypto key generate dsa` command to generate a DSA key pair and enable the SSH server.

**Configuration Examples for Implementing Secure Shell**

This section provides the following configuration example:
Configuring Secure Shell: Example

The following 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
```

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA configuration tasks</td>
<td>Configuring AAA Services on the Cisco ASR 9000 Series Router Software module in Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide.</td>
</tr>
<tr>
<td>Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Host Services and Applications Commands on the Cisco ASR 9000 Series Router module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.</td>
</tr>
<tr>
<td>Note: IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).</td>
<td>Secure Shell Commands on the Cisco ASR 9000 Series Router Software module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.</td>
</tr>
</tbody>
</table>
### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft-ietf-seesh-connect-17.txt</td>
<td>SSH Connection Protocol, July 2003</td>
</tr>
<tr>
<td>Draft-ietf-seesh-architecture-14.txt</td>
<td>SSH Protocol Architecture, July 2003</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
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.

For a complete description of the Public Key Infrastructure (PKI) commands used here, see the Public Key Infrastructure Commands module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.

---

**Feature History for Implementing Secure Socket Layer**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing Secure Socket Layer, page 126
- Information About Implementing Secure Socket Layer, page 126
- How to Implement Secure Socket Layer, page 127
- Configuration Examples for Implementing Secure Socket Layer, page 129
- Additional References, page 130
Prerequisites for Implementing Secure Socket Layer

The following prerequisites are required to implement SSL:

- 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 the Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide.
- Before you can begin using SSL, you must generate either Rivest, Shamir, and Adelman (RSA) or Digital Signature Algorithm (DSA) key pairs, enroll with a CA, and obtain the CA certificate for the router key.
- SSL servers support Advanced Encryption Standard (AES), which has key sizes of 128, 192, and 256 bits.

For more information on the commands required to perform these tasks, see the crypto key generate rsa, crypto key generate dsa, crypto ca enroll, and crypto ca authenticate commands in the Public Key Infrastructure Commands on the Cisco ASR 9000 Series Router module of the Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference.

Information About Implementing Secure Socket Layer

To implement SSL you need to understand the following concept:

Purpose of Certification Authorities

Certification Authorities (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 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. Internet Key Exchange (IKE), an essential component of IPSec, can use digital signatures to scalable authenticate peer devices before setting up security associations (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.

### How to Implement Secure Socket Layer

To configure SSL so that it can be used by any application, such as HTTP server or object request broker (ORB) server, such as HTTP server or XML management agent, perform the task described in the following section.

### Configuring Secure Socket Layer

This task explains how to configure SSL.

#### SUMMARY STEPS

1. `crypto key generate rsa [usage-keys | general-keys] [keypair-label]`
2. `configure`
3. `domain ipv4 host host-name v4address1 [v4address2...v4address8] [unicast | multicast]`
4. `crypto ca trustpoint ca-name`
5. `enrollment url CA-URL`
6. Use the `commit` or `end` command.
7. `RP/0/RSP0/CPU0:router# cryptoca authenticate ca-name`
8. `crypto ca enroll ca-name`
9. `show crypto ca certificates`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> crypto key generate rsa [usage-keys</td>
<td>general-keys] [keypair-label]</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router# crypto key generate</code></td>
<td>• RSA key pairs are used to sign and encrypt Internet Key Exchange (IKE) key management messages and are required before you can obtain a certificate for your router.</td>
</tr>
</tbody>
</table>
### Implementing Secure Socket Layer

**Purpose**
- Use the `usage-keys` keyword to specify special usage keys; use the `general-keys` keyword to specify general-purpose RSA keys.
- The `keypair-label` argument is the RSA key pair label that names the RSA key pairs.
- To generate DSA key pairs, use the `crypto key generate dsa` command in EXEC mode.

**Command or Action**
- `rsa general-keys`

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>`domain ipv4 host host-name v4address1 [v4address2...v4address8] [unicast</td>
<td>multicast]`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# domain ipv4 host ultra5 192.168.7.18</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>crypto ca trustpoint ca-name</code></td>
<td>Configures a trusted point with a selected name so that your router can verify certificates issued to peers.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint myca</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>enrollment url CA-URL</code></td>
<td>Specifies the URL of the CA.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-trustp)# enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Use the <code>commit</code> or <code>end</code> command.</td>
<td><strong>commit</strong>—Saves the configuration changes and remains within the configuration session. <strong>end</strong>—Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# commit</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>crypto ca authenticate ca-name</code></td>
<td>This command authenticates the CA to your router by obtaining the CA certificate, which contains the public key for the CA.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# crypto ca authenticate myca</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto ca authenticate myca</td>
<td>- When prompted, type <strong>y</strong> to accept the certificate.</td>
</tr>
</tbody>
</table>

**Step 8**

**Example:**

```
RP/0/RSP0/CPU0:router# crypto ca authenticate myca
```

**crypto ca enroll** **ca-name**

Requests certificates for all of your RSA key pairs.

- 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.
- Verify that the certificate has been granted by using the `show crypto ca certificates` command.

**Step 9**

**Example:**

```
RP/0/RSP0/CPU0:router# show crypto ca certificates
```

**show crypto ca certificates**

Displays information about your certificate and the CA certificate.

**Configuration Examples for Implementing Secure Socket Layer**

This section provides the following configuration example:

### Configuring Secure Socket Layer: Example

The following example shows how to generate the RSA keys for the router, configure a trust point, authenticate the CA server, obtain a certificate from the CA for the key, and display information about the certificate:

```
crypto key generate rsa general-keys commit configure domain ipv4 host xyz-ultra5 10.0.0.5 crypto ca trustpoint myca enrollment url http://xyz-ultra5 end
crypto ca authenticate myca crypto ca enroll myca show crypto ca certificates
```
Additional References

The following sections provide references related to implementing SSL.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Public Key Infrastructure Commands on the Cisco ASR 9000 Series Router module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
<tr>
<td>SSL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Secure Socket Layer Protocol Commands on the Cisco ASR 9000 Series Router module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
</table>
### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Layer 2 Security Features

This module provides an overview of security features for Layer 2 services. All Layer 2 security features must be configured at the VPLS bridge domain level.

- Security Features for Layer 2 VPLS Bridge Domains, page 133

Security Features for Layer 2 VPLS Bridge Domains

This table lists security features for Layer 2 VPLS bridge domains and points you to the detailed configuration documentation for each feature.

Table 4: Security Features for Layer 2 VPNs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC address-based traffic blocking, filtering, and limiting on VPLS bridge domains</td>
<td>In the <em>Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide</em>, in the module &quot;Implementing Virtual Private LAN Services on Cisco ASR 9000 Series Routers,&quot; see the &quot;Configuring the MAC Address-related Parameters&quot; section.</td>
</tr>
<tr>
<td>Traffic storm control on VPLS bridge domains</td>
<td>In the <em>Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide</em> (this publication), see the module &quot;Implementing Traffic Storm Control under a VPLS Bridge on Cisco ASR 9000 Series Router.&quot;</td>
</tr>
<tr>
<td>DHCP snooping on VPLS bridge domains</td>
<td>In the <em>Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide</em>, see the module &quot;Implementing DHCP on Cisco ASR 9000 Series Routers.&quot; That module describes both DHCP relay services and DHCP snooping at Layer 2.</td>
</tr>
<tr>
<td>Feature</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IGMP snooping on VPLS bridge domains</td>
<td>In the <em>Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide</em>, see the module “Implementing Layer 2 Multicast with IGMP Snooping.”</td>
</tr>
</tbody>
</table>
Implementing Traffic Storm Control under a VPLS Bridge

Traffic storm control provides Layer 2 port security under a Virtual Private LAN Services (VPLS) bridge by preventing excess traffic from disrupting the bridge. This module describes how to implement traffic storm control.

Feature History for Traffic Storm Control

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>Support was added for traffic storm control under a VPLS bridge.</td>
</tr>
</tbody>
</table>
| Release 5.1 | Support was added for:  
| | • configuring storm control at bridge domain level  
| | • allow storm control rate to be configured in kbps instead of pps |

- Prerequisites for Implementing Traffic Storm Control, page 135
- Restrictions for Implementing Traffic Storm Control, page 136
- Information About Implementing Traffic Storm Control, page 136
- How to Configure Traffic Storm Control, page 138
- Configuration Examples for Traffic Storm Control, page 142
- Additional References, page 145

Prerequisites for Implementing Traffic Storm Control

The following prerequisites are required before implementing traffic storm control:

- The network must be configured with a VPLS bridge domain in an MPLS Layer 2 VPN.
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 Traffic Storm Control

In Cisco IOS XR software Release 3.7.0 FCI, the following restrictions apply:

- Traffic storm control is not supported directly on the bridge domain. The feature must be configured on Ethernet flow points (EFPs) under the bridge domain, using bridge domain submodes. The supported submodes are those used for configuring ACs and access PWs.
- Traffic storm control is not supported for aggregated EFPs (bundles).
- Traffic storm control is not supported for forwarding pseudowires (VFI PWs).
- Immediately after an route switch processor (RSP) failover, traffic storm control drop counters might not be accurate. This loss of counter information after a failover is expected behavior for Cisco IOS XR software counters.
- No alarms are generated when packets are dropped.

Information About Implementing Traffic Storm Control

To implement traffic storm control, you should understand the following concepts:

Understanding Traffic Storm Control

A traffic storm occurs when packets flood a VPLS bridge, creating excessive traffic and degrading network performance. Traffic storm control prevents VPLS bridge disruption by suppressing traffic when the number of packets reaches configured threshold levels. You can configure separate threshold levels for different types of traffic on each port under a VPLS bridge.

Traffic storm control monitors incoming traffic levels on a port and drops traffic when the number of packets reaches the configured threshold level during any 1-second interval. The 1-second interval is set in the hardware and is not configurable. The number of packets allowed to pass during the 1-second interval is configurable, per port, per traffic type. During this interval, it compares the traffic level with the traffic storm control level that the customer configures.

When the incoming traffic reaches the traffic storm control level configured on the bridge port, traffic storm control drops traffic until the end of storm control interval.

Traffic storm control level can be configured separately for these traffic types:

- Broadcast Traffic
- Multicast Traffic
- Unknown Unicast Traffic

The thresholds are configured using a packet-per-second (pps) rate. When the number of packets of the specified traffic type reaches the threshold level on a port, the port drops any additional packets of that traffic.
Traffic storm control has little impact on router performance. Packets passing through ports are counted regardless of whether the feature is enabled. Additional counting occurs only for the drop counters, which monitor dropped packets.

No alarms are produced when packets are dropped.

---

**Traffic Storm Control Defaults**

- The traffic storm control feature is disabled by default. It must be explicitly enabled on each port for each traffic type.
- The traffic storm control monitoring interval is set in the hardware and is not configurable. On Cisco ASR 9000 Series Router, the monitoring interval is always 1 second.

**Supported Traffic Types for Traffic Storm Control**

On each VPLS bridge port, you can configure up to three storm control thresholds—one for each of the supported traffic types. If you do not configure a threshold for a traffic type, then traffic storm control is not enabled on that port or interface for that traffic type.

The supported traffic types are:

- Broadcast traffic—Packets with a packet destination MAC address equal to FFFF.FFFF.FFFF.
- Multicast traffic—Packets with a packet destination MAC address not equal to the broadcast address, but with the multicast bit set to 1. The multicast bit is bit 0 of the most significant byte of the MAC address.
- Unknown unicast traffic—Packets with a packet destination MAC address not yet learned.

Traffic storm control does not apply to bridge protocol data unit (BPDU) packets. All BPDU packets are processed as if traffic storm control is not configured.

**Supported Ports for Traffic Storm Control**

In Cisco IOS XR software Release 3.7.0 FCI, you can configure traffic storm control on the following components under a VPLS bridge domain:

- VPLS bridge domain ACs
- VPLS bridge domain access PWs
Traffic Storm Control Thresholds

Traffic storm control thresholds are configured at a packet-per-second rate. A threshold is the number of packets of the specified traffic type that can pass on a port during a 1-second interval. Valid values for traffic storm control thresholds are integers from 1 to 160000. The maximum value would permit about 19 percent of bandwidth to pass per second on a 10-Gbps link, assuming a 1500-byte packet size.

Traffic Storm Control Drop Counters

Traffic storm control counts the number of packets dropped per port and traffic type. The drop counters are cumulative until you explicitly clear them. Use the `show l2vpn bridge-domain detail` and `show l2vpn forwarding detail` commands to see drop counts. Use the `clear l2vpn forwarding counters` command to clear drop counters.

How to Configure Traffic Storm Control

This section describes how to configure traffic storm control:

Enabling Traffic Storm Control on an AC under a Bridge

Perform this task to enable traffic storm control on an AC under a VPLS bridge. The following task shows how to enable traffic storm control on an AC that is a VLAN on an Ethernet interface.

To disable traffic storm control, navigate to the submode you were in when you enabled the feature, and issue the `no` form of the command.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group bridge-group-name
4. bridge-domain bridge-domain-name
5. interface interface-name
6. storm-control {broadcast | multicast | unknown-unicast} pps packet-threshold
7. Use the commit or end command.
8. show l2vpn bridge-domain bd-name bridge-name detail
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CP00:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>l2vpn</td>
<td>Enters L2 VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/0/CP00:router(config)# l2vpn RP/0/0/CP00:router(config-l2vpn)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>bridge group <em>bridge-group-name</em></td>
<td>Enters L2 VPN bridge group configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/0/CP00:router(config-l2vpn)# bridge group cisco RP/0/0/CP00:router(config-l2vpn-bg)#</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>bridge-domain <em>bridge-domain-name</em></td>
<td>Enters L2 VPN bridge domain configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/0/CP00:router(config-l2vpn-bg)# bridge-domain abc RP/0/0/CP00:router(config-l2vpn-bg-bd)#</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>interface <em>interface-name</em></td>
<td>Names an AC under the bridge domain. In this case, the AC is a VLAN on an Ethernet interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/0/CP00:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/1/0/0.100 RP/0/0/CP00:router(config-l2vpn-bg-bd-ac)#</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>storm-control {broadcast</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/0/CP00:router(config-l2vpn-bg-bd-ac)# storm-control broadcast pps 4500 RP/0/0/CP00:router(config-l2vpn-bg-bd-ac)# storm-control multicast pps 500 RP/0/0/CP00:router(config-l2vpn-bg-bd-ac)#</td>
<td></td>
</tr>
</tbody>
</table>
| Step 7 | 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:  
  • Yes—Saves configuration changes and exits the configuration session. |
Enabling Traffic Storm Control on a PW under a Bridge

Perform this task to enable traffic storm control on a pseudowire under a VPLS bridge.

**Note**
To disable traffic storm control, navigate to the submode you were in when you enabled the feature, and issue the `no` form of the command.

### SUMMARY STEPS

1. `configure`
2. `l2vpn`
3. `bridge group bridge-group-name`
4. `bridge-domain bridge-domain-name`
5. `neighbor address pw-id id`
6. `storm-control {broadcast | multicast | unknown-unicast} pps packet-threshold`
7. Use the `commit` or `end` command.
8. `show l2vpn bridge-domain bd-name bridge-name detail`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Step 2</td>
<td>l2vpn</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config)# l2vpn</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn)#</td>
</tr>
<tr>
<td>Step 3</td>
<td>bridge group bridge-group-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn)# bridge group cisco</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg)#</td>
</tr>
<tr>
<td>Step 4</td>
<td>bridge-domain bridge-domain-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg)# bridge-domain abc</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd)#</td>
</tr>
<tr>
<td>Step 5</td>
<td>neighbor address pw-id id</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd)# neighbor 1.1.1.1 pw-id 100</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)#</td>
</tr>
<tr>
<td>Step 6</td>
<td>storm-control {broadcast</td>
</tr>
<tr>
<td></td>
<td>multicast</td>
</tr>
<tr>
<td></td>
<td>unknown-unicast} pps packet-threshold</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)# storm-control broadcast pps 4500</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)# storm-control multicast pps 500</td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)#</td>
</tr>
<tr>
<td>Step 7</td>
<td>Use the commit or end command.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
clearing traffic storm control drop counters

perform this task to reset traffic storm control drop counters to zero.

summary steps

1. clear l2vpn forwarding counters

detailed steps

<table>
<thead>
<tr>
<th>command or action</th>
<th>purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>step 1</td>
<td>clears l2vpn forwarding counters, including storm control drop counters.</td>
</tr>
<tr>
<td>clear l2vpn forwarding counters</td>
<td>clears l2vpn forwarding counters, including storm control drop counters.</td>
</tr>
</tbody>
</table>

configuration examples for traffic storm control

this section includes the following configuration examples:

configuring traffic storm control on an ac: example

the following example shows broadcast and multicast storm control configuration on an ac under a vpls bridge.

rp/0/rsp0/cpu0:router# show run
[lines deleted]
bridge group 215
bridge-domain 215
mtu 9000
Implementing Traffic Storm Control under a VPLS Bridge

Configuring Traffic Storm Control on an Access PW: Example

The following example shows broadcast and multicast storm control configuration on an access PW under a VPLS bridge.

RP/0/RSP0/CPU0:router# show run 12vpn
bridge group bg_storm_pw
bridge-domain bd_storm_pw
interface Bundle-Ether101
neighbor 10.10.30.30 pw-id 1
storm-control unknown-unicast pps 120
storm-control multicast pps 110
storm-control broadcast pps 100
!
!
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain group bg_storm_pw detail
Bridge group: bg_storm_pw, bridge-domain: bd_storm_pw, ID: 2, state: up, ShgId: 0, MSTi: 0
MAC learning: enabled
MAC withdraw: disabled
Flooding:
  Broadcast & Multicast: enabled
  Unknown unicast: enabled
MAC aging time: 300 s, Type: inactivity
MAC limit: 4000, Action: none, Notification: syslog
MAC limit reached: no
Security: disabled
Split Horizon Group: none
DHCPv4 snooping: disabled
IGMP Snooping profile: none
Bridge MTU: 1500
Filter MAC addresses:
  ACs: 1 (1 up), VFIs: 0, PWs: 1 (1 up)
List of ACs:
  AC: Bundle-Ether101, state is up
    Type Ethernet
    MTU 1500; XC ID 0xfffc0003; interworking none
    MAC learning: enabled
    Flooding:
      Broadcast & Multicast: enabled
      Unknown unicast: enabled
    MAC aging time: 300 s, Type: inactivity
    MAC limit: 4000, Action: none, Notification: syslog
    MAC limit reached: no
    Security: disabled
    Split Horizon Group: none
    DHCPv4 snooping: disabled
    IGMP Snooping profile: none
Storm Control: disabled
Static MAC addresses:
Statistics:
  packets: received 0, sent 5205
  bytes: received 0, sent 645420
Storm control drop counters:
  packets: broadcast 0, multicast 0, unknown unicast 0
  bytes: broadcast 0, multicast 0, unknown unicast 0
List of Access PWs:
  PW: neighbor 10.10.30.30, PW ID 1, state is up (established)
    PW class not set, XC ID 0xfffc0006
    Encapsulation MPLS, protocol LDP
    PW type Ethernet, control word disabled, interworking none
    PW backup disable delay 0 sec
    Sequencing not set
PW Status TLV in use
  MPLS
    Local                      Remote
    Label 16001               16001
    Group ID 0x2              0x2
  Interface Access PW Access PW
    MTU 1500                  1500
    Control word disabled     disabled
    PW type Ethernet          Ethernet
    VCCV CV type 0x2          0x2
      (LSP ping verification) (LSP ping verification)
    VCCV CC type 0x6          0x6
      (router alert label)    (router alert label)
      (TTL expiry)            (TTL expiry)
Incoming Status (PW Status TLV):
  Status code: 0x0 (Up) in Notification message
Outgoing Status (PW Status TLV):
  Status code: 0x0 (Up) in Notification message
Create time: 16/12/2008 00:06:08 (01:00:22 ago)
Configuring Traffic Storm Control on the Bridge Domain: Example

This section contains configuration examples for configuring traffic storm control on the bridge domain:

Configuring Storm Control for Broadcast Traffic: Example

This example shows how to configure storm control for broadcast traffic.

```
(config)# l2vpn
(config-l2vpn)# bridge group grp
(config-l2vpn-bg)# bridge-domain bd
(config-l2vpn-bg-bd)# storm-control broadcast kbps 770
(config-l2vpn-bg-bd)# commit
```

Configuring Storm Control for Multicast Traffic: Example

This example shows how to configure storm control for multicast traffic.

```
(config)# l2vpn
(config-l2vpn)# bridge group grp
(config-l2vpn-bg)# bridge-domain bd
(config-l2vpn-bg-bd)# storm-control multicast pps 88
(config-l2vpn-bg-bd)# commit
```

Configuring Storm Control for Unknown-Unicast Traffic: Example

This example shows how to configure storm control for unknown-unicast traffic.

```
(config)# l2vpn
(config-l2vpn)# bridge group grp
(config-l2vpn-bg)# bridge-domain bd
(config-l2vpn-bg-bd)# storm-control unknown-unicast kbps 1280
(config-l2vpn-bg-bd)# commit
```

Additional References

For additional information related to implementing traffic storm control, refer to the following references.
## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS Layer 2 VPNs</td>
<td>Implementing MPLS Layer 2 VPNs on Cisco ASR 9000 Series Router module in the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide.</td>
</tr>
<tr>
<td>MPLS VPLS bridges</td>
<td>Implementing Virtual Private LAN Services on Cisco ASR 9000 Series Router module in the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide.</td>
</tr>
<tr>
<td>Getting started material</td>
<td>Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide</td>
</tr>
</tbody>
</table>

## Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

1 Not all supported standards are listed.

## MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>No new or modified RFCs are supported, and support for existing RFCs has not been modified.</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
INDEX

A
AAA (authentication, authorization, and accounting) 2, 4, 6, 7, 8, 11, 15, 18, 20, 21, 23, 28, 31, 33, 35, 37, 39, 43, 45, 46, 48, 49, 50
accounting services, enabling 48
authorization, enabling 46
configuring 2, 7, 18, 20, 21, 23, 31, 33, 35, 37, 39, 43, 49, 50
AAA service restrictions 2
AAA service restrictions 2
accounting method lists 43
authentication method lists 37
authorization method lists 39
individual users 21
login parameters 49
RADIUS server groups 33
remote AAA 7
router to RADIUS server communication 23
services (examples) 50
TACACS+ server 31
TACACS+ server groups, 35
task groups for task-based authorization 18
user groups 20
database 6
interim accounting records, generating 45
interim accounting records, procedure 45
per VRF (VPN routing and forwarding) 28
per VRF (VPN routing and forwarding) definition 28
router to RADIUS server communication, configuring 23
task-based authorization 11
task IDs 11
user and group attributes 2
user groups 4, 15
definition 4
inheritance 4
predefined 4
privilege level mapping as an alternative to task IDs 15
XML schema 15
aaa accounting command 45
aaa accounting update command 45
AAA service restrictions 2
accept-lifetime command 76
accounting method lists 43
accounting records, interim 45
procedure 45
accounting services, enabling 48
Additional References command 51, 67, 81, 105, 123, 130, 145
authenticating 62
authenticating the CA 62
authentication 8
authentication method lists 37
authentication option 70
authorization method lists 39
authorization, enabling 46

B
benefits 98
broadcast traffic, traffic storm control support 137

C
CA description 55
Call Interception 86
CAs 57
implementing with 57
implementing without 57
CAs (certification authorities) 54, 55, 57, 58, 60, 62, 63, 126
authenticating 62
declaring 60
description 55, 126
domain names, configuring (example) 57
host names 57
manual enrollment, how to cut-and-paste 63
RSA (Rivest, Shamir, and Adelman) key pairs 58

See also certificates\ 55, 126
supported standards 54
trusted point, configuring 60
certificates 54, 62
requests 62
certificates (continued)
See also CAs\ 62
certification authority interoperability 54, 55, 57, 58, 60, 62, 63, 126
authenticating the CA 62
CA description 55
configuring 57, 60
  domain names (example) 57
  host names (examples) 57
  trusted points 60
description 126
generating RSA (Rivest, Shamir, and Adelman) key pairs 58
manual enrollment, cutting and pasting 63
requesting certificates from the CA 62
See also certificates\ 55
supported standards 54
  Internet Key Exchange (IKE) Security protocol 54
  IP Network Security (IPSec) protocol 54
  Public-Key Cryptography Standard #10 (PKCS#10) 54
  Public-Key Cryptography Standard #7 (PKCS#7) 54
  RSA (Rivest, Shamir, and Adelman) keys 54
  Secure Socket Layer (SSL) protocol 54
  X.509v3 certificate 54

clearing 142
clearing drop counters 142
client 113, 121
  configuring 121
  description 113
  server support 113
clock set command 107
Configuration Example for Inband Management Plane Feature
Enablement command 92
Configuration Examples for Configuring AAA Services
command 50
Configuration Examples for Implementing Certification Authority
Interoperability command 65
Configuration Examples for Implementing Keychain Management
command 80
Configuration Examples for Implementing Management Plane
Protection command 104
Configuration Examples for Implementing Secure Shell
command 122
Configuration Examples for Implementing Secure Socket Layer
command 129
Configuration Examples for Traffic Storm Control command 142
configuring 2, 7, 18, 20, 21, 23, 26, 31, 33, 35, 37, 39, 43, 49, 57, 60, 71,
  73, 74, 77, 117, 121, 127, 138
  AAA service restrictions 2
  accounting method lists 43
  authentication method lists 37
  authorization method lists 39
  dead-server detection 26
  domain names (example) 57
  host names (examples) 57
  individual users 21
configuring (continued)
  key identifiers 73
  key string text 74
  login parameters 49
  outbound traffic 77
  outbound traffic (key chain) 77
  RADIUS server groups 33
  remote AAA 7
  router to RADIUS server communication 23
  TACACS+ server 31
  TACACS+ server groups, 35
task groups for task-based authorization 18
  trusted points 60
  UDP ports 23
  user groups 20
Configuring AAA Services 50
  Example command 50
Configuring Certification Authority Interoperability 65
  Example command 65
configuring cryptographic algorithm 79
Configuring Keychain Management 80
  Example command 80
Configuring Management Plane Protection 104
  Example command 104
Configuring Secure Shell 123
  Example command 123
Configuring Secure Socket Layer 129
  Example command 129
Configuring the Inband Management Plane Protection Feature 92
  Example command 92
Configuring Traffic Storm Control on an AC 142
  Example command 142
Configuring Traffic Storm Control on an Access PW 143
  Example command 143
tcontrol plane protection 97
control plane protection, MPP 97
definition 97
CRLs\ 55, 62, 126

d
Data Interception 86
database 6
dead-server detection 26
  RADIUS 26
    radius-server dead-criteria time command 26
    radius-server dead-criteria tries command 26
deadtime command 33
declaring 60
defaults 137
definition 4, 54, 96, 97
deleting 58
description 55, 70, 95, 97, 113, 125, 126, 138
device configuration 98
device configuration, MPP 98
Disabling Lawful Intercept 89
domain names (example) 57
domain names, configuring (example) 57
domain names, configuring CA interoperability 57
drop counters 138
drop counters, traffic storm control 138, 142
clearing 142
description 138

E

enabling on a PW under a bridge 140
enabling on an AC under a bridge 138
enabling traffic storm control on an AC 138
enabling traffic storm control on an PW 140
end-time 70
end-time, key chain 70
end-time, key chain management 70

F

flooding 136

G

generating 58
generating RSA (Rivest, Shamir, and Adelman) key pairs 58

H

hitless key rollover 72
procedure 72
hitless key rollover, configuring 72
host names 57
host names (examples) 57
host names, configuring CA interoperability (examples) 57

I

Implementing Lawful Intercept, Restrictions 85
implementing with 57
implementing with CAs 57
implementing without 57
implementing without CAs 57
inband 96

inband interface 98
inband management interface, MPP 96
definition 96
individual users 21
inheritance 4
interim accounting records, generating 45
interim accounting records, procedure 45
Internet Key Exchange (IKE) Security protocol 54
IP Network Security (IPSec) protocol 54
IPSec (IP Network Security Protocol) 57
CAs 57
implementing with 57
implementing without 57
IPSec (IPSec Network Security Protocol) 57
implementing with CAs 57
implementing without CAs 57
IPSec\ 55, 126

K

key (key chain) command 73
key chain 70, 71
configuring 71
end-time 70
key chain command 71
lifetime 70
overview 70
start-time 70
key chain command 71
key chain management 70, 71, 73, 74, 76, 77
configuring 71, 73, 74, 77
key identifiers 73
key string text 74
outbound traffic 77
description 70
end-time 70
key lifetime 70
key validation 76
start-time 70
key identifiers 73
key lifetime 70
key string 74
key-string command 74
key string text 74
key validation 76
key-string command 74
keyboard-interactive authentication 117
keys 54, 58
definition 54
deleting 58
L
Lawful Intercept Implementation 85
Lawful Intercept Topology 88
lawful intercept, implementing 83
lifetime 70
lifetime, key chain 70
login parameters 49

M
MAC (message authentication code) 70, 79
authentication option 70
configuring cryptographic algorithm 79
management interface 96
inband 96
out-of-band 96
management plane 97
description 97
MPP feature 97
overview 97
manual enrollment, cutting and pasting 63
manual enrollment, how to cut-and-paste 63
MPP (Management Plane Protection) 95, 96, 97, 98
benefits 98
control plane protection 97
description 95, 97
device configuration 98
management interface 96
inband 96
out-of-band 96
management plane 97
description 97
peer-filtering option 97
MPP feature 97
multicast traffic, traffic storm control support 137

O
operation 17
out-of-band 96
out-of-band interface 101
out-of-band management interface, MPP 96
definition 96
outbound traffic 77
outbound traffic (key chain) 77
overview 70, 97

P
peer keyword 98, 101
inband interface 98
out-of-band interface 101
peer-filtering option 97, 98, 101
definition 97
peer keyword 98, 101
inband interface 98
out-of-band interface 101
per VRF (VPN routing and forwarding) 28
per VRF (VPN routing and forwarding) AAA 28
procedure 28
supported VSAs 28
per VRF (VPN routing and forwarding) definition 28
per VRF AAA 28
predefined 4
prerequisite for traffic storm control 135
prerequisites 112, 126
prerequisites, configuring 112
prerequisites, implementing 126
privilege level mapping as an alternative to task IDs 15
procedure 28, 45, 72
Public-Key Cryptography Standard #10 (PKCS#10) 54
Public-Key Cryptography Standard #7(PKCS#7) 54

R
RADIUS 17, 23, 26
configuring 23, 26
dead-server detection 26
UDP ports 23
operation 17
RADIUS server groups 33
radius-server dead-criteria time command 26
radius-server dead-criteria tries command 26
radius-server deadtime command 26
RAs (registration authorities) 57
RAs[CAs (certification authorities) 55, 126
zzz] 55, 126
remote AAA 7
requesting certificates from the CA 62
requests 62
restrictions 112, 136
restrictions, implementing 112
router to RADIUS server communication 23
router to RADIUS server communication, configuring 23
RSA (Rivest, Shamir, and Adelman) 54, 58
keys 54, 58
definition 54
deleting 58
RSA (Rivest, Shamir, and Adelman) key pairs 58
generating 58
RSA (Rivest, Shamir, and Adelman) keys 54
RSA keys [certificates 62
zzz] 62

S

SAM (Software Authentication Manager) 107
SAM (Software Authentication Manager) description 107
Secure Socket Layer (SSL) protocol 54
See also CAs 62
See also certificates 55, 126
send-lifetime command 77
server 113
server support 113
SFTP (Standard File Transfer Protocol) description 115
show key chain command 71
show radius dead-criteria host command 26
SSH (Secure Shell) 111, 112, 113, 115, 117, 121
cient 113, 121
3DES support 113
configuring 121
description 113
server support 113
configuring 117
prerequisites 112
prerequisites, configuring 112
restrictions 112
restrictions, implementing 112
server 113
SFTP (Standard File Transfer Protocol) description 115
supported versions 111
troubleshooting 121
SSL (Secure Socket Layer) 125, 126, 127
configuring 127
description 125
prerequisites 126
prerequisites, implementing 126
start-time 70
start-time, key chain 70
start-time, key chain management 70
supported ports 137
supported standards 54
Internet Key Exchange (IKE) Security protocol 54
IP Network Security (IPSec) protocol 54
Public-Key Cryptography Standard #10 (PKCS#10) 54
Public-Key Cryptography Standard #7(PKCS#7) 54
RSA (Rivest, Shamir, and Adelman) keys 54
Secure Socket Layer (SSL) protocol 54
X.509v3 certificate 54
supported traffic types 137
supported versions 111

T

TACACS+ server 31
TACACS+ server groups 35
task groups for task-based authorization 18
task IDs 11
task-based authorization 11
task IDs 11
thresholds 138
traffic storm control 136, 137, 138, 140, 142
clearing drop counters 142
configuring 138
defaults 137
drop counters 138
enabling on a PW under a bridge 140
enabling on an AC under a bridge 138
restrictions 136
supported ports 137
supported traffic types 137
thresholds 138
understanding 136
troubleshooting 121
trusted point, configuring 60
trusted points 60

U

UDP ports 23
understanding 136
unicast traffic, traffic storm control support 137
user and group attributes 2
user groups 4, 15, 20
definition 4
inheritance 4
predefined 4
privilege level mapping as an alternative to task IDs 15

V

VPLS bridge 135, 136, 138, 140
enabling traffic storm control on an AC 138
enabling traffic storm control on an PW 140
flooding 136
prerequisite for traffic storm control 135
VSAs (vendor-specific attributes) 28
per VRF AAA 28
supported VSAs 28
Index

X

X.509v3 certificate 54
XML schema 15

Z

zzz] 55, 62, 126