Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 6.1.x

First Published: 2016-08-12

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

CONTENTS

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 *System Security Command Reference for Cisco ASR 9000 Series Routers*.

From Release 6.1.2 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.2 document.

The preface contains the following sections:

Changes to this Document

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

Date	Summary
April 2016	Initial release of this document.

Changes to This Document

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

Table 1: Changes to This Document

Date	Summary
November 2016	Initial release of this document.
February 2016	Added documentation for MACsec feature.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed Feature Information

This chapter lists all the features that have been added or modified in this guide. The table also contains references to these feature documentation sections.

New and Changed System Security Feature Information

Feature	Description	Changed in Release	Where Documented
Implementing MACsec Encryption on the ASR 9000	This feature was modified.	Release 6.0.1	*Implementing MACsec Encryption* chapter. Understanding MACsec Encryption

New and Changed Information

Table 2: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Configuring MACsec Service	This feature is introduced	Release 6.1.2	Types of MACsec Implementation, on page 81 For configuration details, see Implementing MACsec Encryption in the *System Security Configuration guide*. For information on commands, see MACsec Encryption Commands in the *System Security Command Reference*

Feature	Description	Changed in Release	Where Documented
Configuring MACsec Key Rotation with Fallback Pre-Shared Key	This feature is introduced	Release 6.1.3	Types of MACsec Implementation, on page 81For configuration details, see Implementing MACsec Encryption in the *System Security Configuration guide*. For information on commands, see MACsec Encryption Commands in the *System Security Command Reference*

CHAPTER 2

Configuring AAA Services

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

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

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

Note For a complete description of the AAA commands listed in this module, see the *Authentication, Authorization, and Accounting Commands* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*. 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

Release	Modification
Release 3.7.2	This feature was introduced.
Release 4.1.0	Support for VRF aware TACACS+ was added.
Release 6.3.2	Support for Command Accounting was added.

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 14" 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 13 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 5 section.)
- Create, delete, and modify secure domain router users and set user task permissions (after logging in to the secure domain router as the root system user). (See the Secure Domain Router (SDR) Users, on page 5 section.)
- Access fabric racks or any router resource not allocated to a secure domain router, allowing the root system user to authenticate to any router node regardless of the secure domain router configurations.

Root SDR Users

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

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

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

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

Secure Domain Router (SDR) Users

A SDR user has restricted access to an SDR as determined by the root-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 6 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 5 section.)
- Each user task can be assigned read, write, execute, or debug permission.

Predefined User Groups

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

- **cisco-support:** This group is used by the Cisco support team.
- **netadmin:** Has the ability to control and monitor all system and network parameters.
- **operator:** A demonstration group with basic privileges.
- **root-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 5 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 reinherited explicitly.

Task Groups

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

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

Predefined Task Groups

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

- **cisco-support:** Cisco support personnel tasks
- **netadmin:** Network administrator tasks
- **operator:** Operator day-to-day tasks (for demonstration purposes)
- **root-lr:** Secure domain router administrator tasks
- **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 6 section.) For example, when task group A inherits task group B, the new set of attributes of task group A is the union of A and B.

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

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

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

AAA Database

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

Local Database

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

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

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

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

Remote Database

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

Remote AAA Configuration

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

Client Configuration

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

User Groups

User groups that are created in an external server are not related to the user group concept that is used in the context of local AAA database configuration on the router. The management of external TACACS+ server or RADIUS server user groups is independent, and the router does not recognize the user group structure. The remote user or group profiles may contain attributes that specify the groups (defined on the router) to which a user or users belong, as well as individual task IDs. For more information, see the Task IDs for TACACS+ and RADIUS Authenticated Users, on page 14 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 mode).

Authentication

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

Authentication of 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 mode and external API clients can use this feature to optimize their operations. EXEC mode can avoid displaying the commands that are not applicable and an EMS application can, for example, disable graphical user interface (GUI) menus that are not applicable.

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

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:

```
rommon1> AUX_AUTHEN_LEVEL=0
rommon2> sync
rommon2> boot tftp:/ ...
```

Authentication Failure

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

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

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

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

Password Types

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

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

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

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 cannnot 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 3: Task ID Classes

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

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

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

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

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

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

```
RP/0/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 14" 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 16" 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    EXECUTEDEBUG
Task:                 bgp  :READ    WRITE    EXECUTE
Task:                 cdp  :READ
Task:                diag  :READ
Task:          ext-access  :READ             EXECUTE
Task:             logging  :READ
```

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

Example:

```
Username:user2
Password:
RP/0/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 tac plus server, the configuration file has to specify *priv_lvl* in its configuration file, as shown in the following example:

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

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

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

```
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:

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

About RADIUS

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

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

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

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

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

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

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a "smart card" access control system. In one case, RADIUS has been used with Enigma security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco router with RADIUS to the network. This might be the first step when you make a transition to a Terminal Access Controller Access Control System Plus (TACACS+) server.
- Networks in which a user must access only a single service. Using RADIUS, you can control user access to a single host, utility such as Telnet, or protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 10.2.3.4 and the defined access list is started.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions and to efficiently manage the use of shared resources to offer differing service-level agreements.

Network Security Situations in Which RADIUS is Unsuitable

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
    - AppleTalk Remote Access (ARA)
    - NetBIOS Frame Control Protocol (NBFCP)
    - NetWare Asynchronous Services Interface (NASI)
    - X.25 PAD connections

- Router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a router other than a Cisco router if that router requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

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

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
    1. ACCEPT—The user is authenticated.
    1. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
    1. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
    1. 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 mode or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

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

Differentiated Services Code Point (DSCP) Marking support for TACACS packets

Differentiated Services is a Quality of Service (QoS) architecture that manages the data traffic in a network by using the principle of traffic classification. In this model, the traffic is divided into classes and the data packets are forwarded to the corresponding classes. Based on the priority of the network traffic, the different classes are managed.

To classify traffic, Differentiated Services uses Differentiated Services Code Point (DSCP). It is a 6-bit field in the Type of Service (ToS) byte in the IP header. Based on the DSCP value, the user is able to classify the data traffic and forward packets to the next destination.

You can set the value of DSCP. For a single connection, set the DSCP value on the socket while connecting to the server. In this way, all the outgoing packets will have the same DSCP value in their IP headers. For multiple connections, the DSCP value is set on the available open sockets. Use the **tacacs-server ipv4** command to set the DSCP value.

For more information about the DSCP Marking support for TACACS packets feature, see the *Differentiated Services Code Point (DSCP) Marking support for TACACS packets* module in the *System Security*

Configuration Guide for Cisco ASR 9000 Series Routers. For complete command reference of DSCP command, see the *Authentication, Authorization, And Accounting Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**taskgroup** *taskgroup-name* **Example:** `RP/0/RSP0/CPU0:router(config)# taskgroup beta`	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:** `RP/0/RSP0/CPU0:router(config-tg)# description this is a sample task group description`	(Optional) Creates a description of the task group named in Step 2.
Step 4	**task** {**read** | **write** | **execute** | **debug**} *taskid-name* **Example:** `RP/0/RSP0/CPU0:router(config-tg)# task read bgp`	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	**commit**	

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 usergroup that is still referenced in the system results in a warning.

Before you begin

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

SUMMARY STEPS

1. **configure**

2. **usergroup** *usergroup-name*
3. **description** *string*
4. **taskgroup** *taskgroup-name*
5. 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 22.
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**usergroup** *usergroup-name* **Example:** `RP/0/RSP0/CPU0:router(config)# usergroup beta`	Creates a name for a particular user group and enters user group configuration submode. • Specific user groups can be removed from the system by specifying the **no** form of the **usergroup** command.
Step 3	**description** *string* **Example:** `RP/0/RSP0/CPU0:router(config-ug)# description this is a sample user group description`	(Optional) Creates a description of the user group named in Step Step 2, on page 22.
Step 4	**taskgroup** *taskgroup-name* **Example:** `RP/0/RSP0/CPU0:router(config-ug)# taskgroup beta`	Associates the user group named in Step Step 2, on page 23with the task group named in this step. • The user group takes on the configuration attributes (task ID list and permissions) already defined for the entered task group.
Step 5	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 22.	—
Step 6	**commit**	

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 22 section.

Configuring Users

Perform this task to configure a user.

Each user is id entified 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*
    -
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 23.
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**username** *user-name* **Example:** `RP/0/RSP0/CPU0:router(config)# username user1`	RSP0 Creates a name for a new user (or identifies a current user) and enters username configuration submode. • The *user-name* argument can be only one word. Spaces and quotation marks are not allowed.

Step 3	Do one of the following:
	• **password** [**0** | **7**] *password*
	•
	Example:
	`RP/0/RSP0/CPU0:router(config-un)# password 0 pwd1`
	or
	`RP/0/RSP0/CPU0:router(config-un)# secret 0 sec1`
		Specifies a password for the user named in Step 2, on page 23.
		• Use the **secret** command to create a secure login password for the user names specified in Step 2, on page 23.
		• Entering **0** following the **password** command specifies that an unencrypted (clear-text) password follows. Entering **7** following the **password** command specifies that an encrypted password follows.
		•
		• Type **0** is the default for the **password** and **secret** commands.

Step 4	**group** *group-name* **Example:** `RP/0/RSP0/CPU0:router(config-un)# group sysadmin`	Assigns the user named in Step 2, on page 23 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.

	Command or Action	Purpose
Step 5	Repeat Step 4, on page 23 for each user group to be associated with the user specified in Step 2, on page 23.	—
Step 6	**commit**	

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 24 or Configuring a TACACS+ Server, on page 30 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. **commit**
9. show radius

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server host** {*hostname* | *ip address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] **Example:** Specifying a radius server hostname `RP/0/RSP0/CPU0:router(config)# radius-server host host1`	Specifies the hostname or IP address of the remote RADIUS server host. . • 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.

	Command or Action	Purpose
		Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.
Step 3	**radius-server retransmit** *retries* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server retransmit 5`	Specifies the number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up. • In the example, the number of retransmission attempts is set to 5.
Step 4	**radius-server timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server timeout 10`	Sets the number of seconds a router waits for a server host to reply before timing out. • In the example, the interval timer is set to 10 seconds.
Step 5	**radius-server key** {**0** *clear-text-key* | **7** *encrypted-key* | *clear-text-key*} **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server key 0 samplekey`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 6	**radius source-interface** *type instance* [**vrf** *vrf-id*] **Example:** `RP/0/RSP0/CPU0:router(config)# radius source-interface GigabitEthernet 0/3/0/1`	(Optional) Forces RADIUS to use the IP address of a specified interface or subinterface for all outgoing RADIUS 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 RADIUS reverts to the default. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state. The **vrf** keyword enables the specification on a per-VRF basis.
Step 7	Repeat Step 2, on page 25 through Step 6, on page 26 for each external server to be configured.	—
Step 8	**commit**	
Step 9	show radius **Example:** `RP/0/RSP0/CPU0:router# show radius`	(Optional) Displays information about the RADIUS servers that are configured in the system.

What to do next

After configuring router to RADIUS server communication, configure RADIUS server groups. (See the Configuring RADIUS Server Groups, on page 32 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. **commit**
6. **show radius dead-criteria host** *ip address in IPv4 or IPv6 format* [**auth-port** *auth-port*] [**acct-port** *acct-port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	**radius-server deadtime** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server deadtime 5`	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 3	**radius-server dead-criteria time** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria time 5`	Establishes the time for the dead-criteria conditions for a RADIUS server to be marked as dead.
Step 4	**radius-server dead-criteria tries** *tries* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria tries 4`	Establishes the number of tries for the dead-criteria conditions for a RADIUS server to be marked as dead.
Step 5	**commit**	
Step 6	**show radius dead-criteria host** *ip address in IPv4 or IPv6 format* [**auth-port** *auth-port*] [**acct-port** *acct-port*] **Example:** `RP/0/RSP0/CPU0:router# show radius dead-criteria host 172.19.192.80`	(Optional) Displays dead-server-detection information that has been requested for a RADIUS server at the specified IP address.

Configuring Per VRF AAA

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

New Vendor-Specific Attributes (VSAs)

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

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

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

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

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

Table 4: Supported VSAs for Per VRF AAA

VSA Name	Value Type	Description
Note The RADIUS VSAs—rad-serv, rad-serv-source-if, and rad-serv-vrf—must have the prefix "aaa:" before the VSA name.		
rad-serv	string	Indicates the IP address in IPv4 or IPv6 format, 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 "retransmit V," "V" can range from 1 to 100; for "timeout W," the "W" can range from 1 to 1000.
rad-serv-vrf	string	Specifies the name of the VRF that is used to transmit RADIUS packets. The VRF name matches the name that was specified through the **vrf** command.

This task configures RADIUS server groups per VRF. For information about configuring TACACS+ server groups per VRF, refer Configuring TACACS+ Server Groups, on page 34.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server radius** *group-name* **Example:**	Groups different server hosts into distinct lists and enters the server group configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# aaa group server radius radgroup1` `RP/0/RSP0/CPU0:router(config-sg-radius)#`	

Step 3	**server-private** {*hostname* | *ip-address in IPv4 or IPv6 format*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
	Example:
	IP address in IPv4 format
	`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`
	Example:
	IP address in IPv6 format
	`RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 2001:db8:a0b:12f0::1/64 timeout 5`
	`RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 retransmit 3`
		Configures the IP address of the private RADIUS server for the group.
		If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.
		Both **auth-port** and **acct-port** keywords enter RADIUS server-group private configuration mode.

Step 4	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# vrf v2.44.com`	Configures the VRF reference of an AAA RADIUS server group. **Note** Private server IP addresses can overlap with those configured globally and the VRF definitions can help to distinguish them.
Step 5	**commit**	

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 31 for each external server to be configured.

8. **commit**
9. **show tacacs**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**tacacs-server host** *host-name* **port** *port-number*
	Example:
	`RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 port 51`
	`RP/0/RSP0/CPU0:router(config-tacacs-host)#`
		Specifies a TACACS+ host server and optionally specifies a server port number.
		• This option overrides the default, port 49. Valid port numbers range from 1 to 65535.

Step 3	**tacacs-server host** *host-name* **timeout** *seconds*
	Example:
	`RP/0/RSP0/CPU0:router(config-tacacs-host)# tacacs-server host 209.165.200.226 timeout 30`
	`RP/0/RSP0/CPU0:router(config)#`
		Specifies a TACACS+ host server and optionally specifies a timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ server.
		• This option overrides the global timeout value set with the **tacacs-server timeout** command for only this server. The timeout value is expressed as an integer in terms of timeout interval seconds. The range is from 1 to 1000.

Step 4	**tacacs-server host** *host-name* **key** [**0** | **7**] *auth-key* **Example:** `RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 key 0 a_secret`	Specifies a TACACS+ host server and optionally specifies an authentication and encryption key shared between the AAA server and the TACACS+ server. • 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	**tacacs-server host** *host-name* **single-connection** **Example:** `RP/0/RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226 single-connection`	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	**tacacs source-interface** *type instance* **vrf** *vrf-name* **Example:**	(Optional) Specifies the source IP address of a selected interface for all outgoing TACACS+ packets.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# tacacs source-interface GigabitEthernet 0/4/0/0 vrf abc`	• 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 31 for each external server to be configured.	—
Step 8	**commit**	
Step 9	**show tacacs** **Example:** `RP/0/RSP0/CPU0:router# show tacacs`	(Optional) 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 34 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 9 section.)

Before you begin

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

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *group-name*
3. **server** {*hostname* | *ip address in IPv4 or IPv6 format*} [**auth-port** *port-number*] [**acct-port** *port-number*]
4. Repeat Step 4, on page 33 for every external server to be added to the server group named in Step 3, on page 33.
5. **deadtime** *minutes*
6. **commit**
7. **show radius server-groups** [*group-name* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server radius** *group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server radius radgroup1`	Groups different server hosts into distinct lists and enters the server group configuration mode.

Step 3	**server** {*hostname* | *ip address in IPv4 or IPv6 format*} [**auth-port** *port-number*] [**acct-port** *port-number*]
	Example:
	IP address in IPv4 format
	`RP/0/RSP0/CPU0:router(config-sg-radius)# server 192.168.20.0`
	Example:
	IP address in IPv6 format
	`RP/0/RSP0/CPU0:router(config-sg-radius)# server 2001:db8:a0b:12f0::1/64`
		Specifies the hostname or IP address of an external RADIUS server.
		• After the server group is configured, it can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).

Step 4	Repeat Step 4, on page 33 for every external server to be added to the server group named in Step 3, on page 33.	—
Step 5	**deadtime** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# deadtime 1`	Configures the deadtime value at the RADIUS server group level. • The *minutes* 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. The example specifies a one-minute deadtime for RADIUS server group radgroup1 when it has failed to respond to authentication requests for the **deadtime** command **Note** You can configure the group-level deadtime after the group is created.
Step 6	**commit**	
Step 7	**show radius server-groups** [*group-name* [**detail**]] **Example:** `RP/0/RSP0/CPU0:router# show radius server-groups`	(Optional) Displays information about each RADIUS server group that is configured in the system.

What to do next

After configuring RADIUS server groups, define method lists by configuring authentication, authorization, and accounting. (See the Configuring AAA Method Lists, on page 37 section.)

Configuring TACACS+ Server Groups

This task configures TACACS+ server groups.

You can enter one or more **server** commands. The **server** command specifies the hostname or IP address of an external TACACS+ server. Once configured, this server group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting). (See the Method Lists, on page 9 section.)

Before you begin

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

SUMMARY STEPS

1. **configure**
2. **aaa group server tacacs+** *group-name*
3. **server** {*hostname* | *ip address in IPv4 or IPv6 format*}
4. Repeat Step 3, on page 34 for every external server to be added to the server group named in Step 2, on page 34.
5. **server-private** {*hostname* | *ip-address in IPv4 or IPv6 format*} [**port** *port-number*] [**timeout** *seconds*] [**key** *string*]
6. **vrf** *vrf-name*
7. **commit**
8. **show tacacs server-groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server tacacs+** *group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1`	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	**server** {*hostname* | *ip address in IPv4 or IPv6 format*} **Example:** IP address in IPv4 format `RP/0/RSP0/CPU0:router(config-sg-tacacs+)# server 192.168.100.0`	Specifies the hostname or IP address of an external TACACS+ server. • When configured, this group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).
Step 4	Repeat Step 3, on page 34 for every external server to be added to the server group named in Step 2, on page 34.	—
Step 5	**server-private** {*hostname* | *ip-address in IPv4 or IPv6 format*} [**port** *port-number*] [**timeout** *seconds*] [**key** *string*] **Example:**	Configures the IP address of the private TACACS+ server for the group server.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 key a_secret`	**Note** If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.
Step 6	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-tacacs+)# vrf abc`	Specifies the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an AAA TACACS+ server group.
Step 7	**commit**	
Step 8	**show tacacs server-groups** **Example:** `RP/0/RSP0/CPU0:router# show tacacs server-groups`	(Optional) Displays information about each TACACS+ server group that is configured in the system.

What to do next

After configuring TACACS+ server groups, define method lists used by configuring authentication, authorization, and accounting. (See the Configuring AAA Method Lists, on page 37 section.)

Configure Per VRF TACACS+ Server Groups

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

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

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

Prerequisites

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

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

Configuration Example

```
Router#configure
```

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

Router(config)# aaa group server tacacs+ tacgroup1

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

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

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

Running Configuration

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

Verify Per VRF TACACS+ Server Groups

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

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

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

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

Server: 2001:db8:1::2/49 vrf=test-vrf [private]
        opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
```

```
status=up single-connect=false family=IPv6
```

Associated Commands

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

Configuring AAA Method Lists

AAA data may be stored in a variety of data sources. AAA configuration uses *method lists* to define an order of preference for the source of AAA data. AAA may define more than one method list and applications (such as login) can choose one of them. For example, console 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

> **Note** The default method list is applied for all the interfaces for authentication, except when a non-default named method list is explicitly configured, in which case the named method list is applied.
>
> The **group radius, group tacacs+**, and **group** *group-name* forms of the **aaa authentication** command refer to a set of previously defined RADIUS or TACACS+ servers. Use the **radius server-host or tacacs-server host** command to configure the host servers. Use the **aaa group server radius or aaa group server tacacs+** command to create a named group of servers.

SUMMARY STEPS

1. **configure**
2. **aaa authentication** {**login** | **ppp**} {**default** | *list-name* | **remote**} *method-list*
3. **commit**
4. Repeat Step 1 through Step 3 for every authentication method list to be configured.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa authentication** {**login** | **ppp**} {**default** | *list-name* | **remote**} *method-list* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authentication login default group tacacs+`	Creates a series of authentication methods, or a method list. • 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. • Entering the **remote** 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. **Note** The remote keyword is available only on the admin plane. • Entering a *method-list* argument following the method list type. Method list types are entered in the preferred sequence. The listed method types are any one of the following options:

	Command or Action	Purpose
		• **group tacacs+**—Use a server group or TACACS+ servers for authentication • **group radius**—Use a server group or RADIUS servers for authentication • **group** *named-group*—Use a named subset of TACACS+ or RADIUS servers for authentication • **local**—Use a local username or password database for authentication • **line**—Use line password or user group for authentication • The example specifies the **default** method list to be used for authentication.
Step 3	**commit**	
Step 4	Repeat Step 1 through Step 3 for every authentication method list to be configured.	—

What to do next

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

Configuring Authorization Method Lists

This task configures method lists for authorization.

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

Authorization Configuration

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

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

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

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

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

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

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

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

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

- Accept the request as is.
- Refuse authorization.

Creation of a Series of Authorization Methods

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**aaa authorization** {**commands** | **eventmanager** | **exec** | **network**} {**default** | *list-name*} {**none** | **local** | **group** {**tacacs+** | **radius** | *group-name*}}
	Example:
	`RP/0/RSP0/CPU0:router(config)# aaa authorization commands listname1 group tacacs+`
		Creates a series of authorization methods, or a method list.
		• The **commands** keyword configures authorization for all EXEC mode shell commands. Command authorization applies to the EXEC mode commands issued by a user. Command authorization attempts authorization for all EXEC modecommands.
		• The **eventmanager** keyword applies an authorization method for authorizing an event manager (fault manager).
		• The **exec** keyword configures authorization for an interactive (EXEC mode) session.
		• The **network** keyword configures authorization for network services like PPP or IKE.
		• The **default** keyword causes the listed authorization methods that follow this keyword to be the default list of methods for authorization.
		• A *list-name* 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:
		  • **none**—The network access server (NAS) does not request authorization information. Authorization always succeeds. No subsequent authorization methods will be attempted. However, the task ID authorization is always required and cannot be disabled.
		  • **local**—Uses local database for authorization.
		• **group tacacs+**—Uses the list of all configured TACACS+ servers for authorization. The NAS exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating AV

	Command or Action	Purpose
		pairs, which are stored in a database on the TACACS+ security server, with the appropriate user. • **group radius**—Uses the list of all configured RADIUS servers for authorization. • **group** *group-name*—Uses a named server group, a subset of TACACS+ or RADIUS servers for authorization as defined by the **aaa group server tacacs+** or **aaa group server radius** command.
Step 3	**commit**	

What to do next

After configuring authorization method lists, configure accounting method lists. (See the Configuring Accounting Method Lists, on page 42 section.)

Configuring Accounting Method Lists

This task configures method lists for accounting.

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

Accounting Configuration

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

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

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

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

Creation of a Series of Accounting Methods

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	Do one of the following: • **aaa accounting** {**commands** | **exec** | **network**} {**default** | *list-name*} {**start-stop** | **stop-only**} • {none | method} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting commands default stop-only group tacacs+`	**Note** Command accounting is not supported on RADIUS, but supported on TACACS. Creates a series of accounting methods, or a method list. • The **commands** keyword enables accounting for EXEC mode shell commands. • The **exec** keyword enables accounting for an interactive (EXEC mode) session. • The **network** keyword enables accounting for all network-related service requests, such as Point-to-Point Protocol (PPP). • The **default** keyword causes the listed accounting methods that follow this keyword to be the default list of methods for accounting. • A *list-name* character string identifies the accounting method list. • The **start-stop** keyword sends a “start accounting” notice at the beginning of a process and a “stop accounting” notice at the end of a process. The requested user process begins regardless of whether the “start accounting” notice was received by the accounting server. • The **stop-only** keyword sends a “stop accounting” notice at the end of the requested user process. • The **none** keyword states that no accounting is performed.

	Command or Action	Purpose
		• The method list itself follows the **start-stop** keyword. Method list types are entered in the preferred sequence. The method argument lists the following types: • **group tacacs+**—Use the list of all configured TACACS+ servers for accounting. • **group radius—**Use the list of all configured RADIUS servers for accounting. • **group** *group-name*—Use a named server group, a subset of TACACS+ or RADIUS servers for accounting as defined by the **aaa group server tacacs+** or **aaa group server radius** command. • The example defines a **default** command accounting method list, in which accounting services are provided by a TACACS+ security server, with a stop-only restriction.
Step 3	**commit**	

What to do next

After configuring method lists, apply those method lists. (See the Applying Method Lists for Applications, on page 45 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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**aaa accounting update** {**newinfo** | **periodic** *minutes*} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting update periodic 30`	Enables periodic interim accounting records to be sent to the accounting server. • If the **newinfo** 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. • When used with the **periodic** keyword, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all the accounting information recorded for that user up to the time the interim accounting record is sent. **Caution** The **periodic** keyword causes heavy congestion when many users are logged in to the network.
Step 3	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**line** {**aux** | **console** | **default** | **template** *template-name*} **Example:** `RP/0/RSP0/CPU0:router(config)# line console`	Enters line template configuration mode.
Step 3	**authorization** {**commands** | **exec**} {**default** | *list-name*} **Example:** `RP/0/RSP0/CPU0:router(config-line)# authorization commands listname5`	Enables AAA authorization for a specific line or group of lines. • The **commands** keyword enables authorization on the selected lines for all commands. • The **exec** keyword enables authorization for an interactive (EXEC mode) session. • Enter the **default** keyword to apply the name of the default method list, as defined with the **aaa authorization** command. • Enter the name of a list of authorization methods to use. If no list name is specified, the system uses the default. The list is created with the **aaa authorization** command. • The example enables command authorization using the method list named listname5.
Step 4	**commit**	

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 46 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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**line** { **aux** | **console** | **default** | **template** **template-name**} **Example:** `RP/0/RSP0/CPU0:router(config)# line console`	Enters line template configuration mode.
Step 3	**accounting** {**commands** | **exec**} {**default** | *list-name*} **Example:** `RP/0/RSP0/CPU0:router(config-line)# accounting commands listname7`	Enables AAA accounting for a specific line or group of lines. • The **commands** keyword enables accounting on the selected lines for all EXEC mode shell commands. • The **exec** keyword enables accounting for an interactive (EXEC mode) session. • Enter the **default** keyword to apply the name of the default method list, as defined with the **aaa accounting** command. • Enter the name of a list of accounting methods to use. If no list name is specified, the system uses the default. The list is created with the **aaa accounting** command. • The example enables command accounting using the method list named listname7.
Step 4	**commit**	

What to do next

After applying accounting method lists by enabling AAA accounting services, configure login parameters. (See the Configuring Login Parameters, on page 47 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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**line template** *template-name* **Example:** `RP/0/RSP0/CPU0:router(config)# line template alpha`	Specifies a line to configure and enters line template configuration mode.
Step 3	**timeout login response** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-line)# timeout login response 20`	Sets the interval that the server waits for reply to a login. • 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.
Step 4	**commit**	

Configuration Examples for Configuring AAA Services

This section provides the following configuration example:

Configuring AAA Services: Example

The following examples show how to configure AAA services.

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

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

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

```
aaa authentication ppp default local
```

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

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

username user2
secret lab
exit
```

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

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

usergroup uga
taskgroup tga
description usergroup uga
exit
```

Username user2 is configured to inherit from user group uga.

```
username user2
group uga
exit
```

Three TACACS servers are configured.

```
tacacs-server host 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

Related Topic	Document Title
AAA services commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Authentication, Authorization, and Accounting Commands on* the Cisco ASR 9000 Series Router in the

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 3

Implementing Certification Authority Interoperability

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 *System Security Command Reference for Cisco ASR 9000 Series Routers*. 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

Release	Modification
Release 3.7.2	This feature was introduced.

Prerequisites for Implementing Certification Authority

The following prerequisites are required to implement CA interoperability:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must install and activate the Package Installation Envelope (PIE) for the security software.

  For detailed information about optional PIE installation, refer to *System Management Configuration Guide for Cisco ASR 9000 Series Routers*
- You need to have a CA available to your network before you configure this interoperability feature. The CA must support Cisco Systems PKI protocol, the simple certificate enrollment protocol (SCEP) (formerly called certificate enrollment protocol [CEP]).

Restrictions for Implementing Certification Authority

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

Information About Implementing Certification Authority

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

Supported Standards for Certification Authority Interoperability

Cisco supports the following standards:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, a pair of security gateways, or a security gateway and a host.

  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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**hostname** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# hostname myhost`	Configures the hostname of the router.

	Command or Action	Purpose
Step 3	**domain name domain-name** **Example:** `RP/0/RSP0/CPU0:router(config)# domain name mydomain.com`	Configures the IP domain name of the router.
Step 4	**commit**	

Generating an RSA Key Pair

This task generates an RSA key pair.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa** [**usage keys** | **general-keys**] [**keypair-label**] **Example:** `RP/0/RSP0/CPU0:router# crypto key generate rsa general-keys`	Generates RSA key pairs. • 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.
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 you 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

	Command or Action	Purpose
Step 1	**crypto key import authentication rsa** [**usage keys** | **general-keys**] [**keypair-label**] **Example:** `RP/0/RSP0/CPU0:router# crypto key import authentication rsa general-keys`	Generates RSA key pairs. • 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.
Step 2	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.

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**crypto ca trustpoint ca-name** **Example:** `RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint myca`	Declares a CA. • Configures a trusted point with a selected name so that your router can verify certificates issued to peers. • Enters trustpoint configuration mode.
Step 3	**enrollment url CA-URL** **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll`	Specifies the URL of the CA. • The URL should include any nonstandard cgi-bin script location.
Step 4	**query url LDAP-URL** **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com`	(Optional) Specifies the location of the LDAP server if your CA system supports the LDAP protocol.
Step 5	**enrollment retry period minutes** **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# enrollment retry period 2`	(Optional) Specifies a retry period. • After requesting a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a period of time (the retry period) the router will send another certificate request. • Range is from 1 to 60 minutes. Default is 1 minute.
Step 6	**enrollment retry count number** **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# enrollment retry count 10`	(Optional) Specifies how many times the router continues to send unsuccessful certificate requests before giving up. • The range is from 1 to 100.
Step 7	**rsakeypair keypair-label** **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# rsakeypair mykey`	(Optional) Specifies a named RSA key pair generated using the **crypto key generate rsa** command for this trustpoint. • Not setting this key pair means that the trustpoint uses the default RSA key in the current configuration.
Step 8	**commit**	

Authenticating the CA

This task authenticates the CA to your router.

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

SUMMARY STEPS

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

DETAILED STEPS

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

Requesting Your Own Certificates

This task requests certificates from the CA.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ca enroll ca-name** **Example:** `RP/0/RSP0/CPU0:router# crypto ca enroll myca`	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.

	Command or Action	Purpose
		If a timeout occurs, contact your system administrator to get your request approved, and then enter this command again.
Step 2	**show crypto ca certificates** **Example:** `RP/0/RSP0/CPU0:router# show crypto ca certificates`	(Optional) Displays information about the CA certificate.

Configuring Certificate Enrollment Using Cut-and-Paste

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

SUMMARY STEPS

1. **configure**
2. **crypto ca trustpoint** *ca-name*
3. enrollment terminal
4. **commit**
5. **crypto ca authenticate** *ca-name*
6. **crypto ca enroll** *ca-name*
7. **crypto ca import** ca- *name* **certificate**
8. show crypto ca certificates

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**crypto ca trustpoint** *ca-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint myca`
	`RP/0/RSP0/CPU0:router(config-trustp)#`
		Declares the CA that your router should use and enters trustpoint configuration mode.
		• Use the *ca-name* argument to specify the name of the CA.

Step 3	enrollment terminal **Example:** `RP/0/RSP0/CPU0:router(config-trustp)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.
Step 4	**commit**	
Step 5	**crypto ca authenticate** *ca-name* **Example:**	Authenticates the CA by obtaining the certificate of the CA.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# crypto ca authenticate myca`	• Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in Step 2, on page 62.
Step 6	**crypto ca enroll** *ca-name* **Example:** `RP/0/RSP0/CPU0:router# crypto ca enroll myca`	Obtains the certificates for your router from the CA. • Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in Step 2.
Step 7	**crypto ca import** ca- *name* **certificate** **Example:** `RP/0/RSP0/CPU0:router# crypto ca import myca certificate`	Imports a certificate manually at the terminal. • Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in Step 2. **Note** You must enter the **crypto ca import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)
Step 8	show crypto ca certificates **Example:** `RP/0/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.

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

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

crypto key generate rsa mykey

The name for the keys will be:mykey
```

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

show crypto key mypubkey rsa

Key label:mykey
Type     :RSA General purpose
Size     :1024
Created  :17:33:23 UTC Thu Sep 18 2003
Data     :
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CB8D86
 BF6707AA FD7E4F08 A1F70080 B9E6016B 8128004C B477817B BCF35106 BC60B06E
 07A417FD 7979D262 B35465A6 1D3B70D1 36ACAFBD 7F91D5A0 CFB0EE91 B9D52C69
 7CAF89ED F66A6A58 89EEF776 A03916CB 3663FB17 B7DBEBF8 1C54AF7F 293F3004
 C15B08A8 C6965F1E 289DD724 BD40AF59 E90E44D5 7D590000 5C4BEA9D B5020301
 0001

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

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

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

! The following command authenticates the CA to your router.

crypto ca authenticate myca

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

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

crypto ca enroll myca

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

Password:
Re-enter Password:
    Fingerprint: 17D8B38D ED2BDF2E 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. IPSec in the *Implementing IPSec Network Security on* the Cisco ASR 9000 Series Routermodule, and SSL in the *Implementing Secure Socket Layer on* the Cisco ASR 9000 Series Routermodule. These modules are located in *System Security Configuration Guide for Cisco ASR 9000 Series Routers* (this publication).

Additional References

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

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Public Key Infrastructure Commands on* the Cisco ASR 9000 Series Router module in *System Security Command Reference for Cisco ASR 9000 Series Routers*.

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 4

Implementing Keychain Management

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

Feature History for Implementing Keychain Management

Release	Modification
Release 3.7.2	This feature was introduced.

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.

Prior to Cisco IOS XR Release 6.1.4, MACsec protocol does not work when BFD over Bundle (BoB) feature was configured. However, starting from Cisco IOS XR Release 6.1.4 and later releases, MACsec protocol works even if BFD over Bundle (BoB) feature is configured.

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 Comand Reference*.

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

Lifetime of Key

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

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

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

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

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

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

Multiple keychains can be specified.

How to Implement Keychain Management

This section contains the following procedures:

Configuring a Keychain

This task configures a name for the keychain.

You can create or modify the name of the keychain.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **commit**
4. **show key chain** *key-chain-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**key chain** *key-chain-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# key chain isis-keys`
	`RP/0/RSP0/CPU0:router(config-isis-keys)#`
		Creates a name for the keychain.
		Note Configuring only the keychain name without any key identifiers is considered a nonoperation. When you exit the configuration, the router does not prompt you to commit changes until you have configured the key identifier and at least one of the global configuration mode attributes or keychain-key configuration mode attributes (for example, lifetime or key string).

Step 3	**commit**	
Step 4	**show key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router# show key chain isis-keys`	(Optional) Displays the name of the keychain. **Note** The *key-chain-name* argument is optional. If you do not specify a name for the *key-chain-name* argument, all the keychains are displayed.

What to do next

After completing keychain configuration, see the Configuring a Tolerance Specification to Accept Keys, on page 70 section.

Configuring a Tolerance Specification to Accept Keys

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

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **accept-tolerance** *value* [**infinite**]
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# key chain isis-keys`	Creates a name for the keychain.
Step 3	**accept-tolerance** *value* [**infinite**] **Example:** `RP/0/RSP0/CPU0:router(config-isis-keys)# accept-tolerance infinite`	Configures a tolerance value to accept keys for the keychain. • 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.
Step 4	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# key chain isis-keys`	Creates a name for the keychain.
Step 3	**key** *key-id* **Example:** `RP/0/RSP0/CPU0:router(config-isis-keys)# key 8`	Creates a key for the keychain. The key ID number is translated from decimal to hexadecimal to create the command mode subprompt. • Use the *key-id* argument as a 48-bit integer.
Step 4	**commit**	

What to do next

After configuring a key identifier for the keychain, see the Configuring the Text for the Key String, on page 71 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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# key chain isis-keys`	Creates a name for the keychain.

Step 3	**key** *key-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis-keys)# key 8`
	`RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#`
		Creates a key for the keychain.

	Command or Action	Purpose

Step 4	**key-string** [**clear** | **password**] *key-string-text*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# key-string password 8`
		Specifies the text string for the key.
		• Use the **clear** keyword to specify the key string in clear text form; use the **password** keyword to specify the key in encrypted form.
		• For a string to be a valid password, it must comply with the following rules:
		  • It must contain an even number of characters.
		  • The minumum length is 4.
		  • The first two digits must be decimal numbers and the others must be hexadecimal numbers.
		  • The first two digits must not be greater than 53.
		Examples of valid passwords are:
		  • 12abcd
		  • 32986510
		•

Step 5	**commit**	

What to do next

After configuring the text for the key string, see the Configuring the Keys to Generate Authentication Digest for the Outbound Application Traffic, on page 73 section.

Determining the Valid Keys

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# key chain isis-keys`	Creates a a name for the keychain.

Step 3	**key** *key-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis-keys)# key 8`
	`RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#`
		Creates a key for the keychain.

Step 4	**accept-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*] **Example:** `RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# accept-lifetime 1:00:00 october 24 2005 infinite`	(Optional) Specifies the validity of the key lifetime in terms of clock time.
Step 5	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# key chain isis-keys`	Creates a name for the keychain.

	Command or Action	Purpose

Step 3	**key** *key-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis-keys)# key 8`
	`RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#`
		Creates a key for the keychain.

Step 4	**send-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*] **Example:** `RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# send-lifetime 1:00:00 october 24 2005 infinite`	(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 If you intend to set lifetimes on keys, Network Time Protocol (NTP) or some other time synchronization method is recommended.
Step 5	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**key chain** *key-chain-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# key chain isis-keys`
	`RP/0/RSP0/CPU0:router(config-isis-keys)#`
		Creates a name for the keychain.

	Command or Action	Purpose

Step 3	**key** *key-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis-keys)# key 8`
	`RP/0/RSP0/CPU0:router(config-isis-keys-0x8)#`
		Creates a key for the keychain.

Step 4	**cryptographic-algorithm** [**HMAC-MD5** | **HMAC-SHA1-12** | **HMAC-SHA1-20** | **MD5** | **SHA-1**] **Example:** `RP/0/RSP0/CPU0:router(config-isis-keys-0x8)# cryptographic-algorithm MD5`	Specifies the choice of the cryptographic algorithm. You can choose from the following list of algorithms: • HMAC-MD5 • HMAC-SHA1-12 • HMAC-SHA1-20 • MD5 • SHA-1 The routing protocols each support a different set of cryptographic algorithms: • Border Gateway Protocol (BGP) supports only HMAC-MD5 and HMAC-SHA1-12. • Intermediate System-to-Intermediate System (IS-IS) supports HMAC-MD5, SHA-1, MD5. • Open Shortest Path First (OSPF) supports MD5, HMAC-MD5.
Step 5	**commit**	

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

Additional References

The following sections provide references related to implementing keychain management.

Related Documents

Related Topic	Document Title
Keychain management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Keychain Management Commands in the System Security Command Reference for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 5

Configure MACSec

This module describes how to configure Media Access Control Security (MACSec) encryption on the ASR 9000 Series Aggregation Services Routers. MACSec is a Layer 2 IEEE 802.1AE standard for encrypting packets between two MACSec-capable routers.

Feature History for Configure MACSec

Release	Modification
Release 5.3.2	This feature was introduced.
Release 6.0.1	This feature was modified to support VLAN sub-interfaces and bundles.
Release 6.1.2	This feature was modified to introduce MACsec as a service.

Understanding MACsec Encryption

Security breaches can occur at any layer of the OSI model. At Layer 2, some of the common breaches at Layer 2 are MAC address spoofing, ARP spoofing, Denial of Service (DoS) attacks against a DHCP server, and VLAN hopping.

MACsec secures data on physical media, making it impossible for data to be compromised at higher layers. As a result, MACsec encryption takes priority over any other encryption method such as IPsec and SSL, at higher layers. MACsec is configured on Customer Edge (CE) router interfaces that connect to Provider Edge (PE) routers and on all the provider router interfaces.

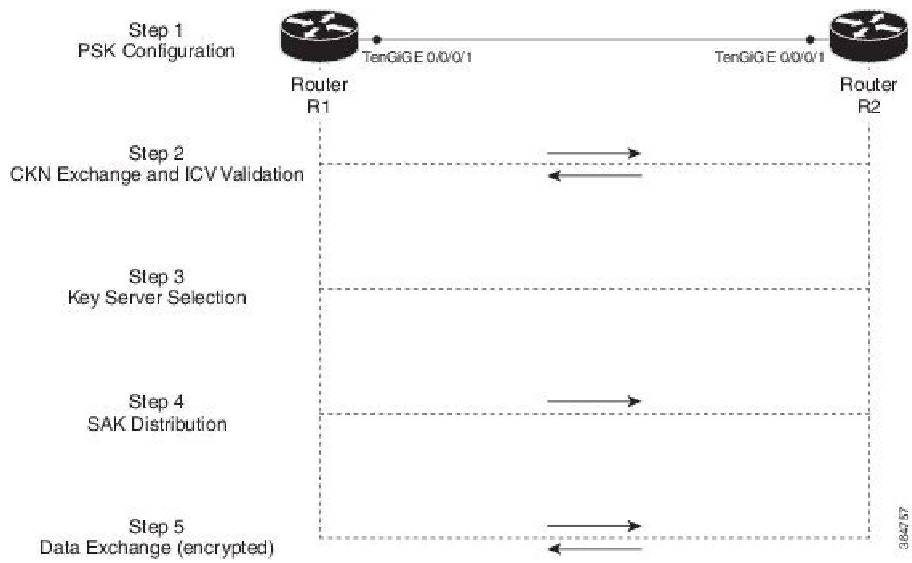
MACservice can be deployed in the network as a technology or as a service. For more information, see Types of MACsec Implementation, on page 81

MACsec Authentication Process

MACsec provides encryption using Advanced Encryption Standard (AES) algorithm at the Layer 2. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

The MACsec encryption process is illustrated in the following figure and description.

Figure 1: MACsec Encryption Process



Step 1: When a link is first established between two routers, they become peers. Mutual peer authentication takes place by configuring a Pre-shared Key (PSK).

Step 2: On successful peer authentication, a connectivity association is formed between the peers, and a secure Connectivity Association Key Name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a Connectivity Association Key (CAK), which is effectively a secret key.

Step 3: A key server is selected between the routers, based on the configured key server priority. Lower the priority value, higher the preference for the router to become the key server. If no value is configured, the default value of 16 is taken to be the key server priority value for the router. Lowest priority value configures that router as the key server, while the other router functions as a key client. The following rules apply to key server selection:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each router selects a peer advertising the highest preference as its key server provided that peer has not selected another router as its key server or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the router with the highest priority SCI is chosen as key server (KS).

Step 4: A security association is formed between the peers. The key server generates and distributes the Secure Association Key (SAK) to the key client (peer). SAKs are generated for every data exchange between the peers.

Step 5: Encrypted data is exchanged between the peers.

Advantages of Using MACsec Encryption

- **Client-Oriented Mode**: MACsec is used in setups where two routers that are peering with each other can alternate as a key server or a key client prior to exchanging keys. The key server generates and maintains the CAK between the two peers.

- **Data Integrity Check**: MACsec uses MKA to generate an Integrity Check Value (ICV) for the frame arriving on the port. If the generated ICV is the same as the ICV in the frame, then the frame is accepted; otherwise it is dropped.

- **Data Encryption**: MACsec provides port-level encryption on the line card of the router. This means that the frames sent out of the configured port are encrypted and frames received on the port are decrypted. MACsec also provides a mechanism where you can configure whether only encrypted frames or all frames (encrypted and plain) are accepted on the interface.
- **Replay Protection**: When frames are transmitted through the network, there is a strong possibility of frames getting out of the ordered sequence. MACsec provides a configurable window that accepts a specified number of out-of-sequence frames.
- **Support for Clear Traffic**: If configured accordingly, data that is not encrypted is allowed to transit through the port.

Types of MACsec Implementation

MACsec is implemented in the following ways:

- **MACsec** where it serves as an encryption method for all traffic on Ethernet links.

  For more information on configuring MACsec, see *Creating a MACsec Keychain* and *Creating a MACsec Policy*

- **MACsec as a service** where it serves as an encryption method for L2VPN and L3VPN traffic over a provider network. It provides a mechanism to provide encryption or decryption service for selected traffic across the WAN core. For example: aservice provider can charge encryption of voice calls at a premium. This solution supports both Point-to-Point as well as Multipoint service for all the traffic on the network.

  For more information on configuring MACsec as a service, see Configuring MACsec as a Service, on page 104

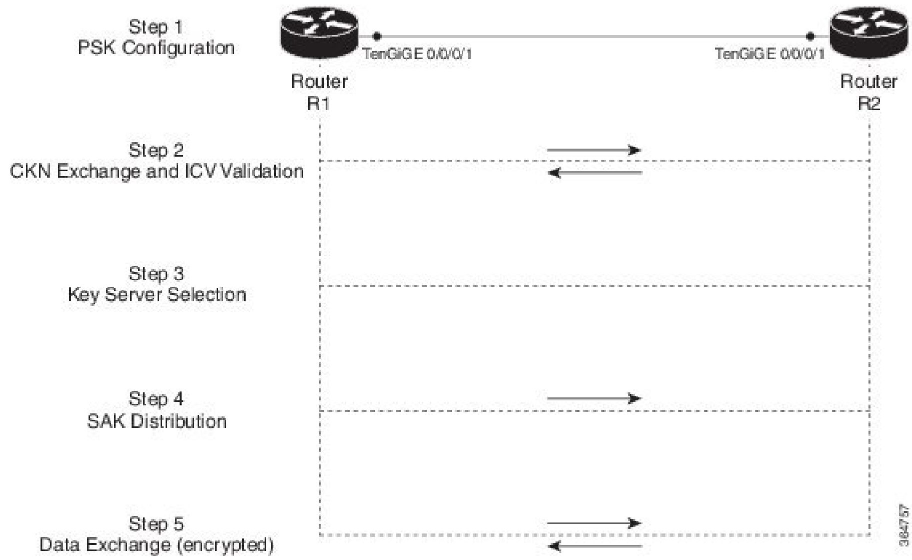
Both MACsec and MACsec service are mutually exclusive and can be deployed in the same network.

MKA Authentication Process

MACsec provides encryption at the Layer 2, which is provided by the Advanced Encryption Standard (AES) algorithm that replaces the DES algorithm. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

The MACsec encryption process is illustrated in the following figure and description.

Figure 2: MKA Encryption Process



Step 1: When a link is first established between two routers, they become peers. Mutual peer authentication takes place by configuring a Pre-shared Key (PSK).

Step 2: On successful peer authentication, a connectivity association is formed between the peers, and a secure Connectivity Association Key Name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a Connectivity Association Key (CAK), which is effectively a secret key.

Step 3: A key server is selected between the routers, based on the configured key server priority. Lower the priority value, higher the preference for the router to become the key server. If no value is configured, the default value of 16 is taken to be the key server priority value for the router. Lowest priority value configures that router as the key server, while the other router functions as a key client. The following rules apply to key server selection:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each router selects a peer advertising the highest preference as its key server provided that peer has not selected another router as its key server or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the router with the highest priority SCI is chosen as key server (KS).

Step 4: A security association is formed between the peers. The key server generates and distributes the Secure Association Key (SAK) to the key client (peer). Each secure channel is supported by an overlapped sequence of Security Associations(SA). Each SA uses a new Secure Association Key (SAK).

Step 5: Encrypted data is exchanged between the peers.

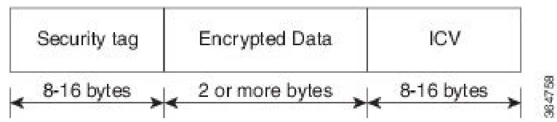
MACsec Frame Format

The MACsec header in a frame consists of three components as illustrated in the following figure.

- **Security tag**: The security tag is 8-16 bytes in length and identifies the SAK to be used for the frame. With Secure Channel Identifier (SCI) encoding, the security tag is 16 bytes in length, and without the encoding, 8 bytes in length (SCI encoding is optional).The security tag also provides replay protection when frames are received out of sequence.

- **Secure data**: This is the data in the frame that is encrypted using MACsec and can be 2 or more octets in length.
- **ICV**: The ICV provides the integrity check for the frame and is usually 8-16 bytes in length, depending on the cipher suite. Frames that do not match the expected ICV are dropped at the port.

Figure 3: MACsec Frame Format



MACSec Support on Line Cards and Routers

MACSec is supported on the following line cards and routers:

- The MACSec technology is supported only on the 200G and 400G modular line cards when used with the following Modular Port Adaptors:
    - 20-port 10 Gigabit Modular Port Adapter (A9K-MPA-20X10GE)
    - 1-port 100 Gigabit Modular Port Adapter (A9K-MPA-1X100GE)
    - 2-port 100 Gigabit Modular Port Adapter (A9K-MPA-2X100GE)
- Cisco ASR 9000 Series 400-Gbps IPoDWDM Line Card (A9K-400G-DWDM-TR)
- 4X100 GE and 8X100 GE OTN Line Card

Note MACSec is not supported on ASR9000 24-port dual-rate 10G/1G service edge–optimized line card (A9K-24X10GE-1G-SE).

Configuring and Verifying MACSec Encryption

MACSec can be configured on physical ethernet interfaces, VLAN sub-interfaces, or interface bundles (link bundles), as explained in this section.

Note MACSec on a VLAN sub-interface is configured in same way as on a physical interface. For a successful MKA session to be up on any VLAN sub-interface, it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet sub-interfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined. The sub-interfaces belonging to a physical interface can have the following encapsulation combinations:

- 802.1Q with a single tag
- 802.1Q with double tags
- 802.1ad with a single tag
- 802.1ad with double tags

Use Case 1: MACSec in a L2VPN

The following figure illustrates the use of MACSec in a L2VPN network. In this topology, MACSec is configured on the PE-facing interfaces of the CE routers. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

In a L2VPN network that uses an Ethernet over MPLS (EoMPLS) pseudowire, the traffic between CE routers is encrypted by MACSec with VLAN tags in clear. The following figure illustrates the use of MACSec in a L2VPN cloud using an EoMPLS pseudowire. MACSec is configured on the PE-facing VLAN sub-interfaces of the CE router. The PE router encapsulates the MACSec frames with VLAN tags and MPLS labels in clear and sends the frames over the EoMPLS pseudowire.

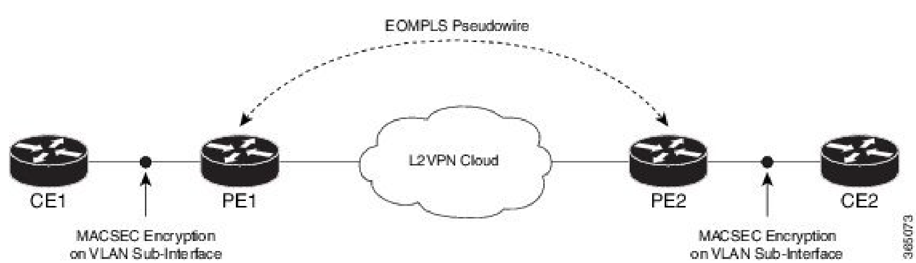
The following table lists the number of sub-interfaces with MACSec supported in a L2VPN.

Note To achieve scaling, sub-interfaces must be used.

Table 5: Supported MACSec Sessions on Sub-Interfaces

Interface Type	No. of Supported MACSec sessions (P2P)
10-GigE	5
40-GigE	21
100-GigE	42

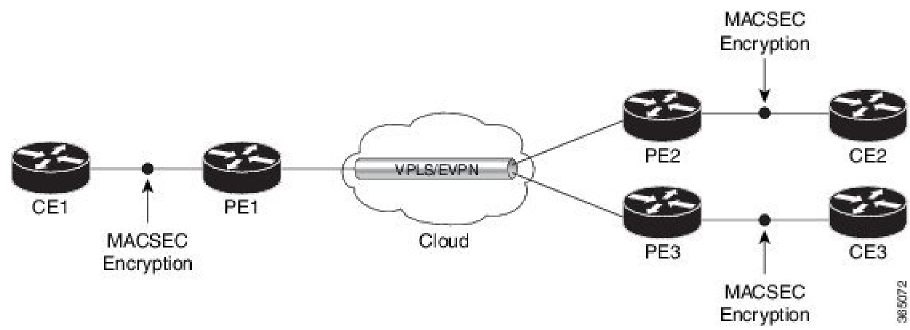
Figure 4: MACSec in a L2VPN Cloud



Use Case 2: MACSec in a VPLS/EVPN

A typical VPLS network often suffers the injection of labeled traffic from potential hackers. The following figure illustrates the use of MACSec in a VPLS/EVPN network for encrypting the data being exchanged over the VPLS cloud. In this topology MACSec is configured on the PE-facing interfaces of the CE routers. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

Figure 5: MACSec in a VPLS/EVPN Cloud



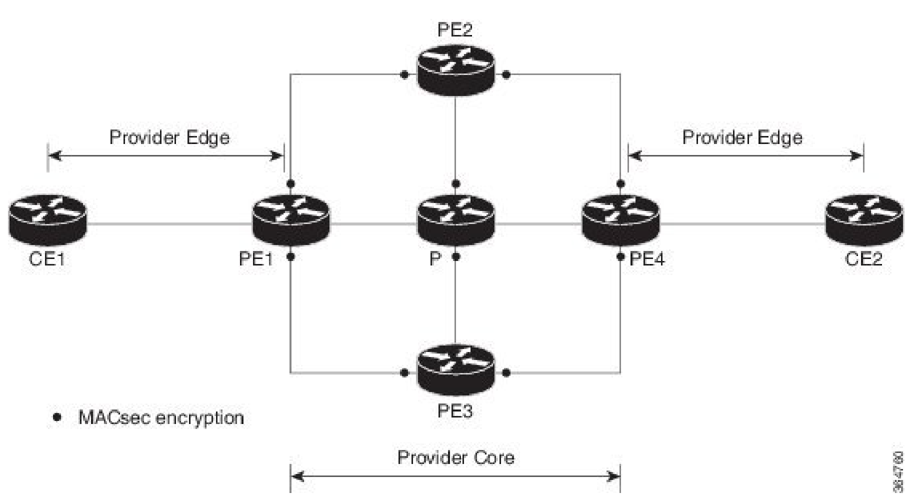
Use Case 3: MACSec in an MPLS Core Network

MACSec in an MPLS core network can be configured on physical interfaces, sub-interfaces or link bundles (Link Aggregation Group or LAG).

In the following topology, MACSec is configured on all router links in the MPLS core. This deployment is useful when the MPLS network spans data centers that are not co-located in the same geography. Each link is, therefore, a link between two data centers and all data exchanged is encrypted using MACSec.

The following figure illustrates the use of MACSec on physical interfaces in an MPLS core network.

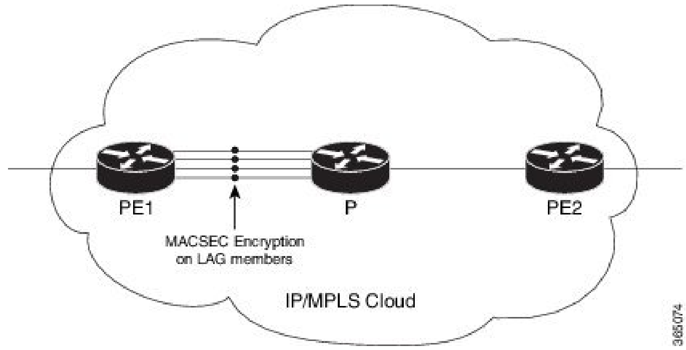
Figure 6: MACSec on Physical Interfaces in an MPLS Core Network



When MACSec is configured on the members of a LAG, an MKA session is set up for each member. SAK is exchanged for each LAG member and encryption/decryption takes place independently of other members in the group. MACSec can also be configured on VLAN sub-interfaces in these networks.

The following figure illustrates the use of MACSec on a link bundle in an MPLS core network.

Figure 7: MACSec on a Link Bundle in an MPLS Core Network



The following section describes procedures for configuring and verifying MACSec configuration in any of the described deployment modes.

Prior to configuring MACSec on a router interface, the MACSec key chain and MACSec policy must be defined. Configuring MACSec encryption involves the following steps:

1. Creating a MACSec Key Chain
2. Creating a MACSec Policy
3. Applying MACSec on a Physical Interface

Creating a MACsec Keychain

A MACsec keychain is a collection of keys used to authenticate peers needing to exchange encrypted information. While creating a keychain, we define the key(s), key string with password, the cryptographic algorithm, and the key lifetime.

MACsec Keychain Keyword	Description
Key	The MACsec key or the CKN can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.
Key-string	The MACsec key-string or the CAK can be either 32 characters or 64 characters in length (32 for AES-128, 64 for AES-256).
Lifetime	This field specifies the validity period of a key. It includes a start time, and an expiry time. We recommend you to set the value for expiry time as *infinite*.

Guidelines for Configuring MACsec Keychain

MACsec keychain management has the following configuration guidelines:

- MKA protocol uses the latest active key available in the Keychain. This key has the latest Start Time from the existing set of currently active keys. You can verify the values using the **show key chain** *keychain-name* command.
- Deletion or expiry of current active key brings down the MKA session resulting in traffic hit. We recommend you to configure the keys with infinite lifetime. If fallback is configured, traffic is safeguarded using fallback on expiry or deletion of primary-keychain active key.
- To achieve successful key rollover (CAK-rollover), the new key should be configured such that it is the latest active key, and kicks-in before the current key expires.
- We recommend an overlap of at least one minute for hitless CAK rollover from current key to new key.
- Start time and Expiry time can be configured with future time stamps, which allows bulk configuration for daily CAK rotation without any intervention of management agent.

SUMMARY STEPS

1. Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac_chain.
2. Provide a name for the MACsec key.
3. Enter the key string and the cryptographic algorithm to be used for the key.
4. Enter the validity period for the MACsec key (CKN) also known as the lifetime period.
5. Commit your configuration.

DETAILED STEPS

Step 1 Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac_chain.

Example:

```
RP/0/RSP0/CPU0:router(config)# keychain mac_chain
```

Step 2 Provide a name for the MACsec key.

The key can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MACsec) # key 1234abcd5678
```

You can also configure a fall-back pre-shared key(PSK) to ensure that a PSK is always available to perform MACsec encryption and decryption. The fallback PSK along with the primary PSK ensures that the session remains active even if the primary PSK is mismatched or there is no active key for the primary PSK.

The configured key is the CKN that is exchanged between the peers.

Note If you are configuring MACsec to interoperate with a MACsec server that is running software prior to Cisco IOS XR Release 6.1.3, then ensure that the MACsec key length is of 64 characters. You can add extra zero characters to the MACseckey so that the length of 64-characters is achieved. If the key length is lesser than 64 characters, authentication will fail.

Step 3 Enter the key string and the cryptographic algorithm to be used for the key.

Example:

The key string is the CAK that is used for ICV validation by the MKA protocol.

```
! For AES 128-bit encryption

RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#
key-string 12345678123456781234567812345678 cryptographic-algorithm AES-128-CMAC

! For AES 256-bit encryption

RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#
key-string 1234567812345678123456781234567812345678123456781234567812345678 cryptographic
-algorithm AES-256-CMAC
```

Note In this example, we have used the AES 256-bit encryption algorithm, and therefore, the key string is 64 hexadecimal characters in length. A 256-bit encryption algorithm uses a larger key that requires more rounds of hacking to be cracked. 256-bit algorithms provide better security against large mass security attacks, and include the security provided by 128-bit algorithms.

Step 4 Enter the validity period for the MACsec key (CKN) also known as the lifetime period.

The lifetime period can be configured, with a duration in seconds, as a validity period between two dates (for example, Jan 01 2014 to Dec 31 2014), or with infinite validity.

The key is valid from the time you configure (in HH:MM:SS format). Duration is configured in seconds.

Example:

```
RP/0/RSP0/CPU0:router(config- mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 01
January 2015 duration 1800
```

An example of configuring the lifetime for a defined period:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20
february 2015 12:00:00 30 september 2015
```

An example of configuring the lifetime as infinite:

```
RP/0/RSP0/CPU0:router(config- mac_chain-MacSec-1234abcd5678)# lifetime
05:00:00 01 January 2015 infinite
```

Note When a key has expired, the MACsec session is torn down and running the **show macsec mka session** command does not display any information. If you run the **show macsec mka interface detail** command, the output displays ***** No Active Keys Present ***** in the PSK information.

Step 5 Commit your configuration.

Example:

```
RP/0/RSP0/CPU0:router(config- mac_chain-MacSec-1234abcd5678)# exit
RP/0/RSP0/CPU0:router (config)# commit
```

This completes the configuration of the MACsec keychain.

Prerequisites for Configuring MACSec on Bundle Member Interfaces

To enable MACSec on bundle members, an user-defined policy must be configured with Should-Secure policy, or Must-Secure policy with **policy-exception LACP-in-clear** command.

Note By default, the system uses the Must-Secure security policy.

Example: Configuring MACSec on Bundle Member With Should-Secure Policy

```
(config)#macsec-policy should-secure
(config-macsec-policy)#security-policy should-secure
(config-macsec-policy)#commit


sh runn macsec-policy should-secure
  macsec-policy should-secure
  security-policy should-secure
!

router(config)# interface HundredGigE 0/1/1/1 # Applying the Should-Secure MACSec Policy
on Bundle Member Interface
  router(config-if)# bundle id 12 mode active
  router(config-if)# macsec psk-keychain kc1 policy should-secure
```

Example: Configuring MACSec on Bundle Member With Must-Secure Policy

```
(config)#macsec-policy must-secure
(config-macsec-policy)#security-policy must-secure
(config-macsec-policy)#policy-exception lacp-in-clear
(config-macsec-policy)#commit

#sh runn macsec-policy must-secure
macsec-policy must-secure
  security-policy must-secure
  policy-exception lacp-in-clear
!
router(config)# interface HundredGigE 0/1/1/2 #Applying the Must-Secure MACSec Policy on
Bundle Member Interface
  router(config-if)# bundle id 12 mode active
  router(config-if)# macsec psk-keychain kc1 policy must-secure
```

Creating a User-Defined MACsec Policy

SUMMARY STEPS

1. Enter the global configuration mode, and enter a name (mac_policy) for the MACsec policy.
2. Configure the cipher suite to be used for MACsec encryption.
3. Configure the confidentiality offset for MACsec encryption.
4. Enter the key server priority.
5. Configure the security policy parameters, either Must-Secure or Should-Secure.

6. Configure the replay protection window size.
7. Configure the ICV for the frame arriving on the port.
8. Commit your configuration and exit the global configuration mode.
9. Confirm the MACsec policy configuration.

DETAILED STEPS

Step 1 Enter the global configuration mode, and enter a name (mac_policy) for the MACsec policy.

Example:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# macsec-policy mac_policy
```

Step 2 Configure the cipher suite to be used for MACsec encryption.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-XPN-256
RP/0/RSP0/CPU0:router(config-mac_policy)#GCM-AES-128
GCM-AES-256
GCM-AES-XPN-128
GCM-AES-XPN-256
```

Note In this example, we have used the GCM-AES-XPN-256 encryption algorithm. A 256-bit encryption algorithm uses a larger key that requires more rounds of hacking to be cracked. 256-bit algorithms provide better security against large mass security attacks, and include the security provided by 128-bit algorithms. Extended Packet Numbering (XPN) is used to reduce the number of key rollovers while data is sent over high speed links. It is therefore highly recommended to use GCM-AES-XPN-256 encryption algorithm for higher data ports.

Step 3 Configure the confidentiality offset for MACsec encryption.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
```

Step 4 Enter the key server priority.

You can enter a value between 0-255. Lower the value, higher the preference to be selected as the key server.

In this example, a value of 0 configures the router as the key server, while the other router functions as a key client. The key server generates and maintains the SAK between the two routers. The default key server priority value is 16.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# key-server-priority 0
```

Step 5 Configure the security policy parameters, either Must-Secure or Should-Secure.

Must-Secure: Must-Secure imposes only MACsec encrypted traffic to flow. Hence, until MKA session is not secured, traffic will be dropped.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy must-secure
```

Should-Secure: Should-Secure allows unencrypted traffic to flow until MKA session is secured. After the MKA session is secured, Should-Secure policy imposes only encrypted traffic to flow.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy should-secure
```

Table 6: MACsec Security Policies

MKA		Secured MKA Session	Unsecured MKA Session
Security Policy	Must-secure	Encrypted traffic	Traffic drop (no Tx and no Rx)
	Should-secure	Encrypted traffic	Plain text or unencrypted traffic

Step 6 Configure the replay protection window size.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# window-size 64
```

This dictates the maximum out-of-sequence frames that are accepted. You can configure a value between 0 and 1024.

Step 7 Configure the ICV for the frame arriving on the port.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# include-icv-indicator
```

This parameter configures inclusion of the optional ICV Indicator as part of the transmitted MACsec Key Agreement PDU (MKPDU). This configuration is necessary for MACsec to interoperate with routers that run software prior to IOS XR version 6.1.3. This configuration is also important in a service provider WAN setup where MACsec interoperates with other vendor MACsec implementations that expect ICV indicator to be present in the MKPDU.

Step 8 Commit your configuration and exit the global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# exit
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:router(config)# exit
```

Step 9 Confirm the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router# show running-config macsec-policy

macsec-policy mac_policy
conf-offset CONF-OFFSET-30
security-policy must-secure
window-size 64
cipher-suite GCM-AES-XPN-256
key-server-priority 0
include-icv-indicator
```

This completes the configuration of the MACsec policy.

Note

- Small packets might be dropped when Data Delay Protection (DDP) is enabled on many MACsec enabled interfaces of a scaled setup. To avoid this, enable DDP only on the interfaces which are absolutely necessary.
- For Cisco ASR 9000 Series Routers to interoperate with Cisco ASR9000 Series Routers that are older than Release 6.2.3, configure a user defined MACsec policy with the `policy-exception lacp-in-clear` command to bring up the MKA sessions over bundle interfaces running in LACP modes.

Applying MACsec Configuration on an Interface

The MACsec service configuration is applied to the host-facing interface of a CE router.

Guidelines for MACsec Interface Configuration

Following are the guidelines for configuring MACsec interface:

- Configure different keychains for primary and fallback PSKs.
- We do not recommend to update both primary and fallback PSKs simultaneously, because fallback PSK is intended to recover MACsec session on primary key mismatch.

SUMMARY STEPS

1. Enter the global configuration mode.
2. Enter the interface configuration mode.
3. Apply the MACsec configuration on an interface.
4. Commit your configuration.

DETAILED STEPS

Step 1 Enter the global configuration mode.

Example:

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

Step 2 Enter the interface configuration mode.

Example:

```
RP/0/RSP0/CPU0:router(config)# interface Te0/3/0/1/4
```

Step 3 Apply the MACsec configuration on an interface.

MACsec PSK Configuration

To apply MACsec PSK configuration on an interface, use the following command.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec psk-keychain mac_chain policy mac_policy
RP/0/RSP0/CPU0:router(config-if)# exit
```

To apply MACsec configuration on a physical interface without the MACsec policy, use the following command.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec psk-keychain script_key_chain2
RP/0/RSP0/CPU0:router(config-if)# exit
```

Step 4 Commit your configuration.

Example:

```
RP/0/RSP0/CPU0:router(config)# commit
```

Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec policy configuration.
2. Verify the MACsec configuration on the respective interface.
3. Verify whether the interface of the router is peering with its neighbor after MACsec configuration
4. Verify whether the MKA session is secured with MACsec on the respective interface.
5. Verify the MACsec session counter statistics.

DETAILED STEPS

Step 1 Verify the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router# show macsec policy mac_policy

============================================================================

Policy      Cipher         Key-Svr     Window  Conf

name        Suite          Priority    Size    Offset

=============================================================================

mac_policy GCM-AES-XPN-256  0          64      30
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

Step 2 Verify the MACsec configuration on the respective interface.

You can verify the MACsec encryption on the configured interface bundle (MPLS network), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka summary

NODE: node0_0_CPU0

==================================================================
Interface     Status    Cipher Suite      KeyChain
==================================================================

Fo0/0/0/1/0   Secured   GCM-AES-XPN-256   mac_chain

Total MACSec Sessions : 1
     Secured Sessions : 1
     Pending Sessions : 0

RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0
===================================================================================
      Interface         Local-TxSCI       # Peers         Status         Key-Server
===================================================================================
    Fo0/0/0/1/0      d46d.5023.3709/0001      1           Secured          YES

! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
=======================================================================
      Interface         Local-TxSCI       # Peers   Status  Key-Server
=======================================================================
  Fo0/0/0/1/1.8      e0ac.f172.4124/001d      1     Secured     Yes
```

The **Status** field in the output confirms that the respective interface is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **Init**.

Note In the VPLS network, because of the configuration on a multi-point interface, the number of live peers displayed is more than 1.

Run the **show run macsec-policy** command in the privileged executive mode to troubleshoot the configuration entered.

Step 3 Verify whether the interface of the router is peering with its neighbor after MACsec configuration

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka session

NODE: node0_0_CPU0

================================================================
Interface    Local-TxSCI           # Peers   Status   Key-Server
================================================================

Fo0/0/0/1/0  001d.e5e9.aa39/0005      1      Secured  YES
```

The **#Peers** field in the output confirms the presence of the peer you have configured on the physical interface, **Fo0/0/0/1/0**. If the number of peers is not reflected accurately in this output, run the **show run** command and verify the peer configuration on the interface.

Note If the MKA session status is shown as **Secured** with **0 (Zero)** peer count, this means that the link is locally secured (Tx). This is because of MKA peer loss caused by **No Rx Packets (MKA Packet)** from that peer.

Step 4 Verify whether the MKA session is secured with MACsec on the respective interface.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : 001d.e5e9.aa39/0005
Local Tx-SSCI           : 1
Interface MAC Address   : 001d.e5e9.aa39
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0
CAK Name (CKN)          : 1020000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A880BB45B9CE01584535F239
Message Number (MN)     : 5382
Authenticator           : NO
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : A880BB45B9CE01584535F23900000001 (1)
Old SAK Status          : FIRST-SAK
Old SAK AN              : 0
Old SAK KI (KN)         : FIRST-SAK (0)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : scale-21
Key Server Priority     : 20
Replay Window Size      : 40
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000001 (GCM-AES-128)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES
# of MACsec Capable Live Peers           : 1
# of MACsec Capable Live Peers Responded : 1
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  4E33A276E7F79C04D80FE346       27114  d46d.5023.3704/0001     2         235
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------


! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/1.8 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4124/001d
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4124
MKA Port Identifier     : 29
Interface Name          : Fo0/0/0/1/1.8
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : 1EC4A4D1B0D75D3D5C2F6393
Message Number (MN)     : 1915
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 3
```

```
Latest SAK KI (KN)       : EB1E04894327E4EFA283C66200000003 (3)
Old SAK Status           : No Rx, No Tx
Old SAK AN               : 0
Old SAK KI (KN)          : RETIRED (4)
SAK Transmit Wait Time   : 0s (Not waiting for any peers to respond)
SAK Retire Time          : 0s (No Old SAK to retire)
MKA Policy Name          : test12
Key Server Priority      : 0
Replay Window Size       : 1024
Confidentiality Offset   : 50
Algorithm Agility        : 80C201
SAK Cipher Suite         : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability        : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired           : YES
# of MACsec Capable Live Peers           : 1
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
  EB1E04894327E4EFA283C662        1908  001d.e5e9.b1c0/0037     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
RP/0/RSP0/CPU0:macsec-CE1#sh macsec mka  interface Fo0/0/0/1/1.8
==============================================================
Interface-name          KeyChain-Name          Policy Name
==============================================================
Fo0/0/0/1/1.8           kc3                    test12


! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka session interface FortyGigE0/0/0/1/0.1 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI             : e0ac.f172.4123/0001
Local Tx-SSCI            : 1
Interface MAC Address    : e0ac.f172.4123
MKA Port Identifier      : 1
Interface Name           : Fo0/0/0/1/0.1
CAK Name (CKN)           : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)   : A1DB3E42B4A543FBDBC281A6
Message Number (MN)      : 1589
Authenticator            : NO
Key Server               : NO
MKA Cipher Suite         : AES-128-CMAC
Latest SAK Status        : Rx & Tx
Latest SAK AN            : 1
Latest SAK KI (KN)       : AEC899297F5B0BDEF7C9FC6700000002 (2)
Old SAK Status           : No Rx, No Tx
Old SAK AN               : 0
Old SAK KI (KN)          : RETIRED (1)
SAK Transmit Wait Time   : 0s (Not waiting for any peers to respond)
SAK Retire Time          : 0s (No Old SAK to retire)
MKA Policy Name          : mk_xpn1
Key Server Priority      : 0
Replay Window Size       : 1024
Confidentiality Offset   : 50
Algorithm Agility        : 80C201
SAK Cipher Suite         : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability        : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired           : YES
# of MACsec Capable Live Peers           : 2
```

```
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
  AEC899297F5B0BDEF7C9FC67         225  001d.e5e9.b1bf/0001     3           0
  0A4C49EE5B7401F1BECB7E22         147  001d.e5e9.f329/0001     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

Step 5 Verify the MACsec session counter statistics.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0


MKA Statistics for Session on interface (Fo0/0/0/1/0)
=======================================================
Reauthentication Attempts.. 0

CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated.... 0
Group CAKs Received..... 0

SA Statistics
SAKs Generated.......... 3
SAKs Rekeyed............ 2
SAKs Received........... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted...... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success.......... 5425
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error............ 0
MKPDUs Rx Success.......... 4932

MKPDU Failures
   MKPDU Rx Validation (ICV).............. 0
   MKPDU Rx Bad Peer MN................... 0
   MKPDU Rx Non-recent Peerlist MN........ 0
   MKPDU Rx Drop SAKUSE, KN mismatch...... 0
   MKPDU Rx Drop SAKUSE, Rx Not Set....... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.. 0
   MKPDU Rx Drop SAKUSE, AN Not in Use.... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set. 0
```

```
SAK Failures
   SAK Generation.................. 0
   Hash Key Generation............. 0
   SAK Encryption/Wrap............. 0
   SAK Decryption/Unwrap........... 0
```

! If sub-interfaces are configured, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8

MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
=======================================================
Reauthentication Attempts.. 0
CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0
SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 9
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1973
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 1965
      "Distributed SAK".. 9
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1973
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1965
```

! In a VPLS network with a mulitpoint interface, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1

MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
=======================================================
Reauthentication Attempts.. 0
CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0
SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 2
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1608
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 406
      "Distributed SAK".. 2
      "Distributed CAK".. 0
MKA IDB Statistics
```

```
MKPDUs Tx Success.......... 1608
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error............ 0
MKPDUs Rx Success.......... 1802
```

The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any.

This completes the verification of MACsec encryption on the IOS-XR.

Verifying MACsec Encryption on ASR 9000

MACsec encryption on the router hardware can be verified by running relevant commands in the Privileged Executive Mode.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.
2. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

DETAILED STEPS

Step 1 Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0


IDB Details:
if_sname : Fo0/0/0/1/0
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE
Sectag offset : 8
VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
 cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
```

```
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064

! When more than 1 RX SA is configured in P2MP networks, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname                : Fo0/0/0/1/0.1
  if_handle               : 0x2e40
  Replay window size      : 1024
  Local MAC               : e0:ac:f1:72:41:23
  Rx SC Option(s)         : Validate-Frames Replay-Protect
  Tx SC Option(s)         : Protect-Frames Always-Include-SCI
  Security Policy         : MUST SECURE
  Sectag offset           : 8
  VLAN                    : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                          : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)
  Rx SC 1
    Rx SCI                : 001de5e9f3290001
    Peer MAC              : 00:1d:e5:e9:f3:29
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[1]          : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67
  Rx SC 2
    Rx SCI                : 001de5e9b1bf0001
    Peer MAC              : 00:1d:e5:e9:b1:bf
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[1]          : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
  Tx SC
```

```
    Tx SCI                  : e0acf17241230001
    Active AN               : 1
    Old AN                  : 0
    Next PN                 : 1, 1, 0, 0
    SAK Data
      SAK[1]                : ***

      SAK Len               : 32
      HashKey[1]            : ***
      HashKey Len           : 16
      Conf offset           : 50
      Cipher Suite          : GCM-AES-XPN-256
      CtxSalt[1]            : ae ca 99 28 7f 5b 0b de f7 c9 fc 67
```

The **if_handle** field provides the IDB instance location.

The **Replay window size** field displays the configured window size.

The **Security Policy** field displays the configured security policy.

The **Local Mac** field displays the MAC address of the router.

The **Peer Mac** field displays the MAC address of the peer. This confirms that a peer relationship has been formed between the two routers.

Step 2 Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff
```

Step 3 Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
```

```
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

Step 4 Verify the MACsec Secure Channel (SC) information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0


MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbfffffb8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
      Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
      Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

Configuring and Verifying MACsec Encryption as a Service

This section describes how MACsec can be implemented as a service in a L2VPN or L3VPN setup.

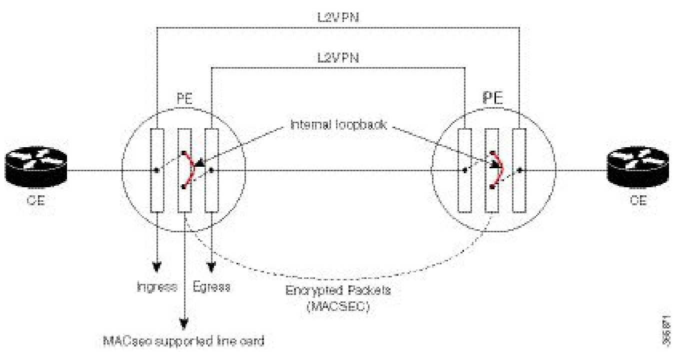
Note MACsec encryption is not supported on interface bundles, but is supported on member links .

Use Case 1: MACsec in an L2VPN Topology

In this topology, MACsec is configured on the PE router (with the interfaces facing the CE router) to provide crypto or encryption service on the PE router as a premium service for selected traffic on the WAN core. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces. The customer can select the traffic that will be part of the encryption.

The following figure illustrates the use of MACsec as a service in an L2VPN network:

Figure 8: MACsec in an L2VPN topology



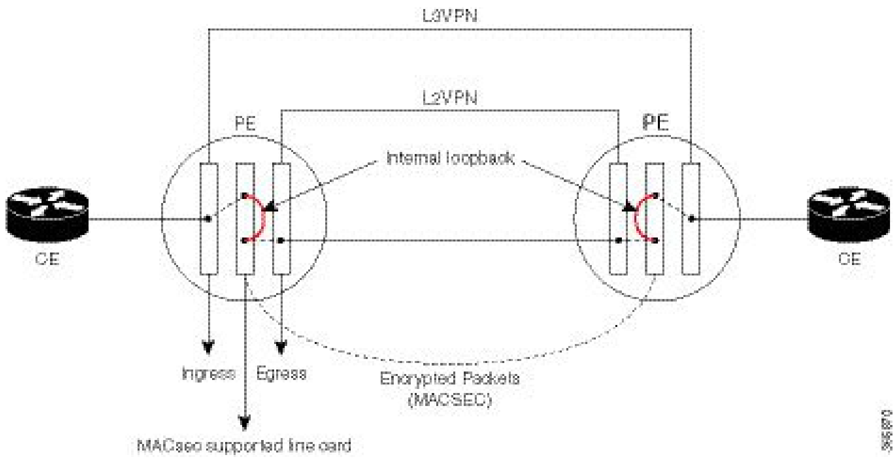
The data transferred between the CE router and the PE router are not encrypted. The data in clear format is sent to the access port of the PE router.

The PE router ports that receive traffic from CE routers divert the traffic using L2 local switching to the line card configured to perform encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE. After this, the packets are sent completely encrypted to the opposite PE router.

Use Case 2: MACsec in an L3VPN Topology

The following figure illustrates the use of MACsec as a service in an L3VPN environment. The topology is similar to an L2VPN set up where MACsec is configured on the PE router (where the interfaces facing the CE router) to provide crypto or encryption services on the PE router as a premium service for selected traffic on the WAN core.

Figure 9:



The data transferred between the CE router and the PE router is not encrypted. The data is sent in clear-text format to the PE router access port. The PE router for each sub-interface distinguishes whether the data is part of MACsec encrypted service.

The PE router ports that receive traffic from CE routers divert the traffic using L3 local switching to the line card port configured to do encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE router. After this, the packets are sent completely encrypted to the opposite PE.

Restrictions

Ports usage for encryption on the line card must meet the following criteria:

- The ports must be TenGigE interfaces.
- Both the ports must belong either to an A9K-MPA-20X10GE MPA, or they must be breakout interfaces from one of the A9K-8X100GE-SE, A9K-8X100GE-TR, A9K-4X100GE-SE, or A9K-4X100GE-TR line cards.
- If the interfaces belong to A9K-MPA-20x10GE line card, then both the interfaces must be either in port range 0-9, or in port range 10-19. One interface from range 0-9 and other from 10-19 must not be selected.
- If the interfaces are breakout interfaces, then both of them must belong to the same HundredGigE port.

Note These restrictions apply only to MACsec interfaces. These restrictions do not apply to the CE or core-facing interfaces.

Configuring MACsec as a Service

SUMMARY STEPS

1. Enter interface configuration mode.
2. Configure the MACsec service.

3. Commit your configuration and exit global configuration mode.
4. Confirm the MACsec policy configuration.

DETAILED STEPS

Step 1 Enter interface configuration mode.

Example:

```
RP/0/RSP/CPU0:router# interface <interface> 15.10 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10
```

Step 2 Configure the MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-subif)# macsec-service decrypt-port <intf>17.10 psk-keychain
<keychain_name> [policy <macsec_policy>]
```

Step 3 Commit your configuration and exit global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router# commit
RP/0/RSP0/CPU0:router# exit
```

Step 4 Confirm the MACsec policy configuration.

Example:

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

show running-config interface <interface> 15.10

interface <interface> 15.10
 macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
 encapsulation dot1q 10
```

Configuring MACsec Service for L2VPN Network

Configuring the MACsec service for L2VPN network, involves the following steps:

SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router.
3. Enable MACsec service.
4. Configure service port.
5. Configure the Xconnect group between ports.

6. Connect the ports.

DETAILED STEPS

Step 1 Enter global configuration mode.

Example:

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

Step 2 Enter interface configuration mode and configure port facing the CE router.

The interface can be a physical interface or a VLAN sub-interface.

Example:

```
RP/0/RSP0/CPU0:router(config)# interface  <interface>15.10  l2transport
 encapsulation dot1q 10
```

Step 3 Enable MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface  <interface>16.10  l2transport
 encapsulation dot1q 10
  macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
```

Step 4 Configure service port.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface  <interface>17.10  l2transport
 encapsulation dot1q 10
```

Step 5 Configure the Xconnect group between ports.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# l2vpn
xconnect group local_macsec
  p2p local_macsec
   interface  <interface>15.10
   interface  <interface>16.10
```

Step 6 Connect the ports.

Example:

```
RP/0/RSP0/CPU0:router(config-if)l2vpn
 xconnect group ext_macsec
  p2p ext_macsec
   interface  <interface>17.10
   neighbor  ipv4  <a.b.c.d>  pw-id  <num>
!
```

Configuring MACsec Service for L3VPN Network

Configuring the MACsec service for L3VPN network, involves the following steps:

SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router
3. Configure the PE1 router with virtual routing details.
4. Enable MACsec service.
5. Configure service port.
6. Configure the Xconnect between ports.
7. Configure ports.
8. Configure OSPF on the core interface.
9. Configure MPLS on the core interface.

DETAILED STEPS

Step 1 Enter global configuration mode.

Example:

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

Step 2 Enter interface configuration mode and configure port facing the CE router

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/4/0/0.1
ipv4 address 161.1.1.1 255.255.255.0
encapsulation dot1q 1
```

Step 3 Configure the PE1 router with virtual routing details.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/1.1
 vrf vrf_1
 ipv4 address 161.1.1.2 255.255.255.0
 encapsulation dot1q 1
```

Step 4 Enable MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/2.1
 vrf vrf_1
 ipv4 address 181.1.1.1 255.255.255.0
 macsec-service decrypt-port TenGigE0/3/0/0/3.1 psk-keychain script_key_chain1
 encapsulation dot1q 1
```

Step 5 Configure service port.

Example:

```
RP/0/RSP0/CPU0:router(config-if)#interface TenGigE0/3/0/0/3.1 l2transport
 encapsulation dot1q 1
!
```

Step 6 Configure the Xconnect between ports.

Example:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#l2vpn
 xconnect group l3serv_xc_gp_1
  p2p l3serv_xc_p2p_1
   interface TenGigE0/3/0/0/3.1
   neighbor ipv4 3.3.3.3 pw-id 1
   !
  !
```

Step 7 Configure ports.

Example:

```
RP/0/RSP0/CPU0:router#(config)
router bgp 100
 bgp router-id 2.2.2.2
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 3.3.3.3
  remote-as 100
  update-source Loopback1
  address-family vpnv4 unicast
  !
 !
 vrf vrf_1
  rd 1234:1
  address-family ipv4 unicast
   redistribute connected
   redistribute static
  !
  neighbor 181.1.1.2
   remote-as 100
   address-family ipv4 unicast
   !
  !
 !
```

Step 8 Configure OSPF on the core interface.

Example:

```
RP/0/RSP0/CPU0:router#
macsec-PE1#sh run router ospf
router ospf core
 router-id 2.2.2.2
 redistribute connected
 redistribute static
 area 0
  interface Loopback1
  !
```

```
  interface TenGigE0/1/0/1
  !
 !
```

Step 9 Configure MPLS on the core interface.

Example:

```
RP/0/RSP0/CPU0:router#
mpls ldp
 graceful-restart
 router-id 2.2.2.2
 interface TenGigE0/1/0/1
 !
!
```

Applying MACsec Service Configuration on an Interface

The MACsec service configuration is applied to the host-facing interface of a CE router.

SUMMARY STEPS

1. Enter the global configuration mode.
2. Enter the interface configuration mode.
3. If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.
4. Apply the MACsec service configuration on an interface.
5. Commit your configuration.

DETAILED STEPS

Step 1 Enter the global configuration mode.

Example:

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

Step 2 Enter the interface configuration mode.

The interface can be a physical interface or a VLAN sub-interface.

Example:

```
RP/0/RSP0/CPU0:router(config)# interface Te0/3/0/1/4
```

Step 3 If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.

Example:

```
! For 802.1q encapsulation with a single tag
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 5

! For 802.1q encapsulation with double tags
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 3 second-dot1q 4

! For 802.1ad encapsulation with a single tag
```

```
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 5

! For 802.1ad encapsulation with double tags
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 3 dot1ad 4
```

Step 4 Apply the MACsec service configuration on an interface.

To apply MACsec service configuration on an interface, use the following configuration.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec-service decrypt-port TenGigE0/3/0/1/5 psk-keychain
script_key_chain1 policy mk_xpn_1tag
RP/0/RSP0/CPU0:router(config-if)# exit
```

Step 5 Commit your configuration.

Example:

```
RP/0/RSP0/CPU0:router(config)# commit
```

Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec policy configuration.
2. Verify the MACsec configuration on the respective interface.
3. Verify whether the interface of the router is peering with its neighbor after MACsec configuration
4. Verify whether the MKA session is secured with MACsec on the respective interface.
5. Verify the MACsec session counter statistics.

DETAILED STEPS

Step 1 Verify the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router# show macsec policy mac_policy

=============================================================================

Policy      Cipher          Key-Svr     Window  Conf

name        Suite           Priority    Size    Offset

==============================================================================

mac_policy GCM-AES-XPN-256  0           64      30
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

Step 2 Verify the MACsec configuration on the respective interface.

You can verify the MACsec encryption on the configured interface bundle (MPLS network), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka summary

NODE: node0_0_CPU0

==================================================================
Interface     Status    Cipher Suite      KeyChain
==================================================================

Fo0/0/0/1/0   Secured   GCM-AES-XPN-256   mac_chain

Total MACSec Sessions : 1
     Secured Sessions : 1
     Pending Sessions : 0

RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0
===================================================================================
      Interface         Local-TxSCI       # Peers         Status         Key-Server
===================================================================================
    Fo0/0/0/1/0      d46d.5023.3709/0001      1           Secured          YES

! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
=======================================================================
      Interface         Local-TxSCI       # Peers   Status  Key-Server
=======================================================================
  Fo0/0/0/1/1.8      e0ac.f172.4124/001d      1     Secured     Yes
```

The **Status** field in the output confirms that the respective interface is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **Init**.

Note In the VPLS network, because of the configuration on a multi-point interface, the number of live peers displayed is more than 1.

Run the **show run macsec-policy** command in the privileged executive mode to troubleshoot the configuration entered.

Step 3 Verify whether the interface of the router is peering with its neighbor after MACsec configuration

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka session

NODE: node0_0_CPU0

================================================================
Interface    Local-TxSCI           # Peers   Status  Key-Server
================================================================
```

```
Fo0/0/0/1/0  001d.e5e9.aa39/0005      1      Secured  YES
```

The **#Peers** field in the output confirms the presence of the peer you have configured on the physical interface, **Fo0/0/0/1/0**. If the number of peers is not reflected accurately in this output, run the **show run** command and verify the peer configuration on the interface.

Note If the MKA session status is shown as **Secured** with **0 (Zero)** peer count, this means that the link is locally secured (Tx). This is because of MKA peer loss caused by **No Rx Packets (MKA Packet)** from that peer.

Step 4 Verify whether the MKA session is secured with MACsec on the respective interface.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : 001d.e5e9.aa39/0005
Local Tx-SSCI           : 1
Interface MAC Address   : 001d.e5e9.aa39
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0
CAK Name (CKN)          : 1020000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A880BB45B9CE01584535F239
Message Number (MN)     : 5382
Authenticator           : NO
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : A880BB45B9CE01584535F23900000001 (1)
Old SAK Status          : FIRST-SAK
Old SAK AN              : 0
Old SAK KI (KN)         : FIRST-SAK (0)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : scale-21
Key Server Priority     : 20
Replay Window Size      : 40
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000001 (GCM-AES-128)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES
# of MACsec Capable Live Peers           : 1
# of MACsec Capable Live Peers Responded : 1
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  4E33A276E7F79C04D80FE346       27114  d46d.5023.3704/0001     2         235
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------

! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/1.8 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4124/001d
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4124
```

```
MKA Port Identifier     : 29
Interface Name          : Fo0/0/0/1/1.8
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : 1EC4A4D1B0D75D3D5C2F6393
Message Number (MN)     : 1915
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 3
Latest SAK KI (KN)      : EB1E04894327E4EFA283C66200000003 (3)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (4)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : test12
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES
# of MACsec Capable Live Peers           : 1
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
  EB1E04894327E4EFA283C662        1908  001d.e5e9.b1c0/0037     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ---------------------------------------------------------------------------
RP/0/RSP0/CPU0:macsec-CE1#sh macsec mka  interface Fo0/0/0/1/1.8
=============================================================
Interface-name          KeyChain-Name           Policy Name
=============================================================
Fo0/0/0/1/1.8           kc3                     test12
```

```
! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka session interface FortyGigE0/0/0/1/0.1 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4123/0001
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4123
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0.1
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A1DB3E42B4A543FBDBC281A6
Message Number (MN)     : 1589
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 1
Latest SAK KI (KN)      : AEC899297F5B0BDEF7C9FC6700000002 (2)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (1)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
```

```
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : mk_xpn1
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES
# of MACsec Capable Live Peers           : 2
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  AEC899297F5B0BDEF7C9FC67         225  001d.e5e9.b1bf/0001     3           0
  0A4C49EE5B7401F1BECB7E22         147  001d.e5e9.f329/0001     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

Step 5 Verify the MACsec session counter statistics.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0


MKA Statistics for Session on interface (Fo0/0/0/1/0)
=======================================================
Reauthentication Attempts.. 0

CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated.... 0
Group CAKs Received..... 0

SA Statistics
SAKs Generated.......... 3
SAKs Rekeyed............ 2
SAKs Received........... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted...... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success.......... 5425
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error............ 0
MKPDUs Rx Success.......... 4932
```

```
MKPDU Failures
   MKPDU Rx Validation (ICV)............. 0
   MKPDU Rx Bad Peer MN.................. 0
   MKPDU Rx Non-recent Peerlist MN........ 0
   MKPDU Rx Drop SAKUSE, KN mismatch...... 0
   MKPDU Rx Drop SAKUSE, Rx Not Set....... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.. 0
   MKPDU Rx Drop SAKUSE, AN Not in Use.... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set. 0

SAK Failures
   SAK Generation.................. 0
   Hash Key Generation............. 0
   SAK Encryption/Wrap............. 0
   SAK Decryption/Unwrap........... 0
```

! If sub-interfaces are configured, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8

MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
=======================================================
Reauthentication Attempts.. 0
CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0
SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 9
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1973
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 1965
      "Distributed SAK".. 9
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1973
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1965
```

! In a VPLS network with a mulitpoint interface, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1

MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
=======================================================
Reauthentication Attempts.. 0
CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0
SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
```

```
   SAKs Received........... 2
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1608
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 406
      "Distributed SAK".. 2
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1608
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1802
```

The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any.

This completes the verification of MACsec encryption on the IOS-XR.

Verifying MACsec Encryption on ASR 9000

MACsec encryption on the router hardware can be verified by running relevant commands in the Privileged Executive Mode.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.
2. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

DETAILED STEPS

Step 1 Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0


IDB Details:
if_sname : Fo0/0/0/1/0
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE
Sectag offset : 8
```

```
VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
 cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064

! When more than 1 RX SA is configured in P2MP networks, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname                : Fo0/0/0/1/0.1
  if_handle               : 0x2e40
  Replay window size      : 1024
  Local MAC               : e0:ac:f1:72:41:23
  Rx SC Option(s)         : Validate-Frames Replay-Protect
  Tx SC Option(s)         : Protect-Frames Always-Include-SCI
  Security Policy         : MUST SECURE
  Sectag offset           : 8
  VLAN                    : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                          : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)
  Rx SC 1
    Rx SCI                : 001de5e9f3290001
    Peer MAC              : 00:1d:e5:e9:f3:29
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[1]          : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67
  Rx SC 2
    Rx SCI                : 001de5e9b1bf0001
    Peer MAC              : 00:1d:e5:e9:b1:bf
    Stale                 : NO
```

```
    SAK Data
      SAK[1]             : ***

      SAK Len            : 32
      HashKey[1]         : ***
      HashKey Len        : 16
      Conf offset        : 50
      Cipher Suite       : GCM-AES-XPN-256
      CtxSalt[1]         : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
  Tx SC
    Tx SCI               : e0acf17241230001
    Active AN            : 1
    Old AN               : 0
    Next PN              : 1, 1, 0, 0
    SAK Data
      SAK[1]             : ***

      SAK Len            : 32
      HashKey[1]         : ***
      HashKey Len        : 16
      Conf offset        : 50
      Cipher Suite       : GCM-AES-XPN-256
      CtxSalt[1]         : ae ca 99 28 7f 5b 0b de f7 c9 fc 67
```

The **if_handle** field provides the IDB instance location.

The **Replay window size** field displays the configured window size.

The **Security Policy** field displays the configured security policy.

The **Local Mac** field displays the MAC address of the router.

The **Peer Mac** field displays the MAC address of the peer. This confirms that a peer relationship has been formed between the two routers.

Step 2 Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff
```

Step 3 Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

Step 4 Verify the MACsec Secure Channel (SC) information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0


MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbfffffb8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
      Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
      Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

CHAPTER 6

Implementing Lawful Intercept

Lawful Intercept is not a part of the Cisco IOS XR software by default. You have to install it separately by installing and activating **asr9k-li-px.pie** .

For more information about activating and deactivating the Lawful Intercept package, see the *Installing and Activating the PIE* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of Lawful Intercept commands, see the *Lawful Intercept Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

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 *System Management Command Reference for Cisco ASR 9000 Series Routers*. To implement lawful intercept, you must understand how the SNMP server functions. For this reason, carefully review the information described in the module *Implementing SNMP* in *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.
- **Lawful intercept must be explicitly disabled—**It is automatically enabled on a provisioned router. However, you should not disable LI if there is an active tap in progress, because this deletes the tap.
- **Management plane configured to enable SNMPv3—**Allows the management plane to accept SNMP commands, so that the commands go to the interface (preferably, a loopback) on the router. This allows the mediation device (MD) to communicate with a physical interface.
- **VACM views enabled for SNMP server—**View-based access control model (VACM) views must be enabled on the router.
- **Provisioned MD—**For detailed information, see the vendor documentation associated with your MD. For a list of MD equipment suppliers preferred by Cisco, see http://www.cisco.com/en/US/tech/tk583/tk799/tsd_technology_support_protocol_home.html.
- **VoIP surveillance-specific requirements**
  - **Lawful-intercept-enabled call agent—**A lawful-intercept-enabled call agent must support interfaces for communications with the MD, for the target of interest to provide signaling information to the MD. The MD extracts source and destination IP addresses and Real-Time Protocol (RTP) port numbers from the Session Description Protocol (SDP) signaling information for the target of interest. It uses these to form an SNMPv3 SET, which is sent to the router acting as the content IAP to provision the intercept for the target of interest.

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

    The MD uses the CISCO-IP-TAP-MIB to set up the filter for the IP addresses and port numbers to be intercepted and derived from the SDP.
  - Routers to be used for calls by the target number must be provisioned for this purpose through the MD.
  - The MD that has been provisioned with the target number to be intercepted.
- **Data session surveillance-specific requirements**
  - Routers to be used by the data target that have been provisioned for this purpose through the MD.
  - **The MD that has been provisioned with the user login ID, mac address of the user CPE device, or the DSLAM physical location ID—**The IP address is the binding that is most frequently used to identify the target in the network. However, alternative forms of information that uniquely identify the target in the network might be used in some network architectures. Such alternatives include the MAC address and the acct-session-id.
- The MD can be located anywhere in the network but must be reachable from the content IAP router, which is being used to intercept the target. MD should be reachable ONLY from global routing table and NOT from VRF routing table.

Restrictions for Implementing Lawful Intercept

The following restrictions are applicable for Lawful Intercept:

- If lawful intercept is set up separately for two inter-communicating hosts with two different mediation devices, then by default, only the ingress traffic on the ASR 9000 router from one of the hosts is intercepted. You can configure the **overlap-tap enable** command to separately intercept the ASR 9000 ingress as well as egress traffic for both the mediation devices.
- 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 Intercept on gigabit ethernet LCs
  - IPv6 MD encapsulation
  - Per layer 3 interface tapping

  Note Per layer 2 interface tapping is supported.

  - 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.

- Lawful intercept is applied only on IP traffic. If IP traffic is configured to leave an egress port as MPLS-encapsulated frames, then lawful intercept will not apply to the egress port, even if overlapping taps are configured.

Information About Lawful Intercept Implementation

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

Lawful intercept offers these capabilities:

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

Interception Mode

The lawful intercept has two interception modes:

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

Overlapping Taps

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

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

Provisioning for VoIP Calls

Lawful Intercept provisioning for VoIP occurs in these ways:

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

Call Interception

VoIP calls are intercepted in this manner:

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

Provisioning for Data Sessions

Provisioning for data sessions occurs in a similar way to the way it does for lawful intercept for VoIP calls. (See Provisioning for VoIP Calls, on page 124.)

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

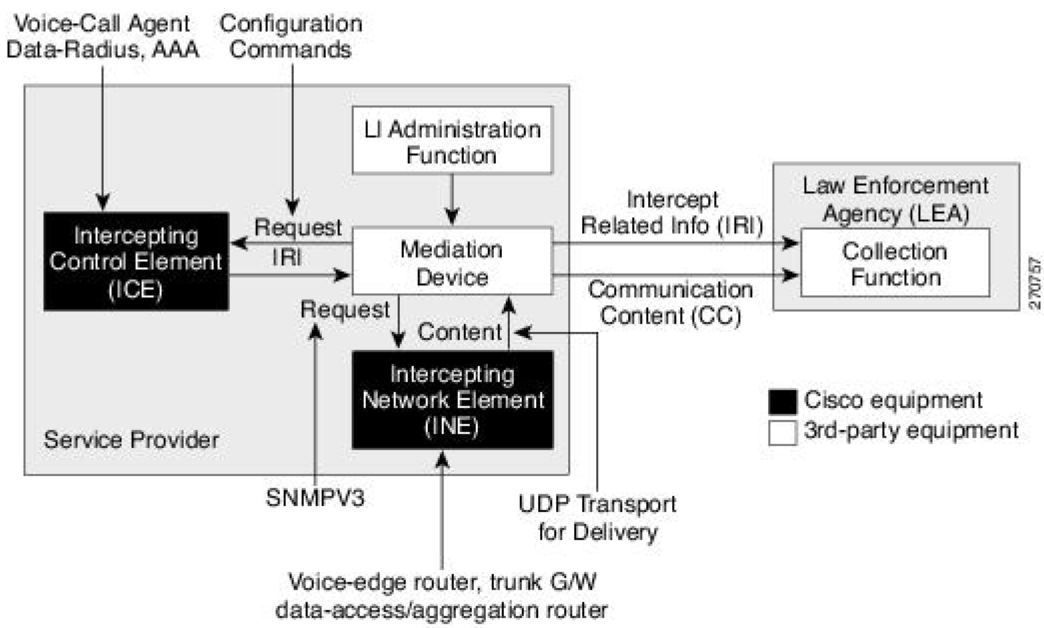
The MD performs these tasks:

- 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 10: Lawful Intercept Topology for Both Voice and Data Interception



Layer 2 Lawful Intercept

You can configure SNMP-based lawful intercept on a layer 2 interface. This intercepts all traffic passing through the particular interface.

For information on Layer 2 lawful intercept, refer Layer 2 Lawful Intercept section.

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 except for the A9K-8x100G-LB-SE and A9K-8x100G-LB-TR line cards. These line cards have a tap limit of 2000 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.
    - 500 Mbps for Modular Weapon-X line cards.
    - 1000 Mbps for 100GE line cards.

- Support upto 512 MDs.

Intercepting IPv4 and IPv6 Packets

This section provides details for intercepting IPv4 and IPv6 packets supported on the Cisco ASR 9000 Series Router.

Lawful Intercept Filters

The filters used for classifying a tap are:

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

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

To further extend filteration criteria for IPv6 packets, an additional support to intercept IPv6 packets based on flow ID has been introduced on the Cisco ASR 9000 Series Router. All IPv6 packets are intercepted based on the fields in the IPv6 header which comprises numerous fields defined in IPv6 Header Field Details table:

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

Table 7: IPv6 Header Field Details

IPv6 Field Name	Field Description	Field Length
Version	IPv6 version number.	4 bits
Traffic Class	Internet traffic priority delivery value.	8 bits
Flow ID (Flow Label)	Used for specifying special router handling from source to destination(s) for a sequence of packets.	20 bits

IPv6 Field Name	Field Description	Field Length
Payload Length	Specifies the length of the data in the packet. When cleared to zero, the option is a hop-by-hop Jumbo payload.	16 bits unassigned
Next Header	Specifies the next encapsulated protocol. The values are compatible with those specified for the IPv4 protocol field.	8 bits
Hop Limit	For each router that forwards the packet, the hop limit is decremented by 1. When the hop limit field reaches zero, the packet is discarded. This replaces the TTL field in the IPv4 header that was originally intended to be used as a time based hop limit.	8 bits unsigned
Source Address	The IPv6 address of the sending node.	16 bytes
Destination Address	The IPv6 address of the destination node.	16 bytes

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

Intercepting VRF (6VPE) and 6PE Packets

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

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

Cisco ASR 9000 Series Router supports intercepting IPv6 packets of the specified VRF ID for 6VPE. To distiguish traffic on VPN, VRFs are defined containing a specific VRF ID. The filter criteria to tap a particular VRF ID is specified in the tap. IPv6 packets are intercepted with the VRF context on both scenarios: imposition (ip2mpls) and disposition (mpls2ip).

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

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

Encapsulation Type Supported for Intercepted Packets

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

The intercepted packet gets a new UDP header and IPv4 header. Information for IPv4 header is derived from MD configuration. Apart from the IP and UDP headers, a 4 byte channel identifier (CCCID) is also inserted after the UDP header in the packet. After adding the MD encapsulation, if the packet size is above the MTU, the egress LC CPU fragments the packet. Moreover, there is a possibility that the packet tapped is already a fragment. Each tap is associated with only one MD. Cisco ASR 9000 Series Router does not support forwarding replicated packets to multiple MDs.

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

Per Tap Drop Counter Support

Cisco ASR 9000 Series Router line cards provide SNMP server as an interface to export each tap forwarded to MD packet and drop counts. Any intercepted packets that are dropped prior to getting forwarded to the MD due to policer action are counted and reported. The drops due to policer action are the only drops that are counted under per tap drop counters. If a lawful intercept filter is modified, the packet counts are reset to 0.

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

High Availability for Lawful Intercept

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

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

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

Preserving TAP and MD Tables during RP Fail Over

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

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

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

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

Replay Timer

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

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

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

Installing Lawful Intercept (LI) Package

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

Installing and Activating the LI Package

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

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

To install the Lawful Intercept (LI) package, you must install and activate the **asr9k-li-px.pie**

For more information about installing PIEs, refer to *Upgrading and Managing Cisco IOS XR Software section* of the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

	Command or Action	Purpose
Step 2	**install add** *tftp://<IP address of tftp server>/<location of pie on server>* **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://172.201.11.140/auto/tftp-users1/asr9k-li-px.pie`	Copies the contents of a package installation envelope (PIE) file to a storage device.
Step 3	**install activate** *device:package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-li-px.pie`	Activates the respective package and adds more functionality to the existing software.
Step 4	**install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits from the admin mode.
Step 6	**show install committed** **Example:** `RP/0/RSP0/CPU0:router# show install committed`	Shows the list of the committed software packages.

Deactivating the LI PIE

To uninstall the Lawful Intercept package, deactivate asr9k-li-px.pie as shown in the following steps:

SUMMARY STEPS

1. **admin**
2. **install deactivate** *device:package*
3. **install commit**
4. **install remove** *device:package*
5. **exit**
6. **show install committed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

	Command or Action	Purpose
Step 2	**install deactivate** *device:package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k-li-px.pie`	Activates the respective package and adds more functionality to the existing software.
Step 3	**install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 4	**install remove** *device:package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-li-px.pie`	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits from the admin mode.
Step 6	**show install committed** **Example:** `RP/0/RSP0/CPU0:router# show install committed`	Shows the list of the committed software packages.

Upgrade and Downgrade Scenarios for the Lawful Intercept package

This section describes the possible upgrade and downgrade scenarios with respect to the Lawful Intercept (LI) package.

This example configuration demonstrates how to upgrade or downgrade the Cisco IOS XR software with or without the LI package. Suppose you have two versions of software images, V1 and V2. If you want to upgrade or downgrade from V1 to V2 without the LI package, you need to perform the following steps for the upgrade or the downgrade procedure:

Note Ensure that you use Turbo Boot to load the image for the downgrade process.

1. Ensure that the device has booted with the V1 image. Check the Package Installation Envelope (PIE) files that have been installed in V1.
2. Save all the PIE files that exist in V2 in the Trivial File Transfer Protocol (TFTP) server. Copy the contents of the PIE files from the TFTP server by using the **install add** command in the admin mode.

```
RP/0/RSP0/CPU0:router(admin)# install add tar
tftp://223.255.254.254/install/files/pies.tar
```

3. To activate all the PIE files in V2 at once, run the following commands based on the type of upgrade:

At any point during the upgrade or the downgrade process, you can check the progress by using the **show install request** or the **show issu** command.

Some of the conventions that are followed in describing these scenarios are:

- Release 4.3.1 base image: It is the Cisco IOS XR software for Release 4.3.1 that contains Cisco LI by default.
- Release 4.3.2 base image: It is the Cisco IOS XR software for Release 4.3.2 that does not contain Cisco LI by default.
- Separate LI package: It is the LI package that needs to be installed separately for Release 4.3.2 and higher versions.

Table 8: Upgrade Scenarios

Upgrade From	Upgrade To	Result	Supported
Release 4.3.1 base image	Release 4.3.2 base image	Before the upgrade, the LI has to be configured and provisioned completely. After the upgrade to Release 4.3.2 version without the LI package, you cannot configure or provision LI.	Yes
Release 4.3.1 base image	Release 4.3.2 base image with the separate LI package	The Upgrade will reload the router. After the upgrade process completes, you need to reconfigure LI MDs/TAPs from the SNMP server. Also, all the LI configurations made in the earlier version is accepted.	Yes
Release 4.3.2 base image with the separate LI package	Release 4.3.3 base image with the separate LI package	After the upgrade, the LI configuration is not retained.	Yes
Release 4.3.2 base image with the separate LI package	Release 4.3.3 base image without the separate LI package	This upgrade is not possible as the installation process will not proceed without the LI PIE.	No
Release 4.3.2 base image without the separate LI package	Release 4.3.3 base image with the separate LI package	This upgrade is possible.	Yes
ISSU for Release 4.3.1 base image	Release 4.3.2 with the separate LI package	After this upgrade, to retain the LI configuration, you have to replay the configuration before the replay timeout occurs.	Yes

Upgrade From	Upgrade To	Result	Supported
ISSU for Release 4.3.2 base image with the separate LI package	Release 4.3.3 with the separate LI package	After this upgrade, to retain the LI configuration, you have to replay the configuration before the replay timeout occurs.	Yes

Table 9: Downgrade Scenarios

Downgrade From	Downgrade To	Result	Supported
Release 4.3.2 base image without the separate LI package	Release 4.3.1 base image	After the downgrade, begin the provisioning process of LI.	Yes
Release 4.3.2 base image with the separate LI package	Release 4.3.1 base image	This scenario is not supported.	No
Release 4.3.3 base image with the separate LI package	Release 4.3.2 base image with the separate LI package	After the downgrade, the LI configuration is not retained. You have to provision the LI once again.	Yes
Release 4.3.3 base image with the separate LI package	Release 4.3.2 base image without the LI package	After the downgrade, the LI configuration is lost. You will not be able to provision it after downgrade.	Yes
Release 4.3.3 base image	Release 4.3.2 base image with the separate LI package	The LI configuration is accepted and can be provisioned only after the downgrade.	Yes
ISSU			No

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

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

Disabling SNMP-based Lawful Intercept

Lawful Intercept is enabled by default on the Cisco ASR 9000 Series Router after installing and activating the **asr9k-li-px.pie**.

- 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/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# lawful-intercept disable
```

Note The **lawful-intercept disable** command is available on the router, only after installing and activating the **asr9k-li-px.pie**.

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 144.

Related Tasks

- Configuring a Device for Management Plane Protection for an Inband Interface, on page 146

Related Examples

- Configuring the Inband Management Plane Protection Feature: Example, on page 139

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. **commit**
8. show snmp users
9. show snmp group
10. show snmp view

DETAILED STEPS

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

	Command or Action	Purpose
Step 6	**snmp-server user** *mduser-id groupname* **v3 auth md5** md-*password* **Example:** `RP/0//CPU0:router(config)# snmp-server mduser-id TapGroup v3 auth md5 mdpassword`	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. • The *mduser-id* and *mdpassword* must match that configured on MD. Alternatively, these values must match those in use on the router. • Passwords must be eight characters or longer to comply with SNMPv3 security minimums. • Minimum Lawful Intercept security level is auth; The noauth option will not work, as it indicates noAuthnoPriv security level. The Lawful Intercept security level must also match that of the MD. • Choices other than MD5 are available on the router, but the MD values must match. Most MDs default to or support only MD5.
Step 7	**commit**	
Step 8	show snmp users **Example:** `RP/0//CPU0:router# show snmp users`	Displays information about each SNMP username in the SNMP user table.
Step 9	show snmp group **Example:** `RP/0//CPU0:router# show snmp group`	Displays information about each SNMP group on the network.
Step 10	show snmp view **Example:** `RP/0//CPU0:router# show snmp view`	Displays information about the configured views, including the associated MIB view family name, storage type, and status.

Adding MD and TAP Objects

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

- cTap2MediationDestAddressType
- cTap2MediationDestAddress
- cTap2MediationDestPort
- cTap2MediationSrcInteraface
- cTap2MediationTimeout

- cTap2MediationTransport
- cTap2MediationStatus

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	Add MD. **Example:** `setany -v3 <ip-address> <user> cTap2MediationDestAddressType.1 <ipv4/ipv6> cTap2MediationDestAddress.1 <"ip"> cTap2MediationDestPort.1 "1234" cTap2MediationSrcInterface.1 0 cTap2MediationTransport.1 udp cTap2MediationNotificationEnable.1 true cTap2MediationTimeout.1 '7 de 6 14 3 4 5 6 2d 1 2' cTap2MediationStatus.1 createAndGo cTap2MediationDestAddressType.1 = ipv4(1) cTap2MediationDestAddress.1 = 46 01 01 02 cTap2MediationDestPort.1 = 1234 cTap2MediationSrcInterface.1 = 0 cTap2MediationTransport.1 = udp(1) cTap2MediationNotificationEnable.1 = true(1) cTap2MediationTimeout.1 = 2014-Jun-20,03:04:05.6,-1:2 cTap2MediationStatus.1 = createAndGo(4)`	Creates an MD for mediation services. To delete a MD, run: `setany -v3 <ip-address> <user> cTap2MediationStatus.1 6 cTap2MediationStatus.1 = destroy(6)`
Step 2	Add TAP. **Example:** `setany -v3 <ip-address> <user> citapStreamInterface.1.1 0 citapStreamAddrType.1.1 <ipv4/ipv6> citapStreamSourceAddress.1.1 "5a 1 1 2" citapStreamSourceLength.1.1 32 citapStreamStatus.1.1 citapStreamInterface.1.1 = 0 citapStreamAddrType.1.1 = ipv4(1) citapStreamSourceAddress.1.1 = 5a 01 01 02 citapStreamSourceLength.1.1 = 32 citapStreamStatus.1.1 = createAndGo(4)`	Creates a TAP for stream operation. To delete a TAP, run: `setany -v3 <ip-address> <user> citapStreamStatus.1.1 6 cTap2StreamStatus.1.1 6 citapStreamStatus.1.1 = destroy(6) cTap2StreamStatus.1.1 = destroy(6)`
Step 3	Activate TAP. **Example:** `setany -v3 <ip-address> <user> cTap2StreamType.1.1`	Activates the TAP for stream operation.

	Command or Action	Purpose

	`ip cTap2StreamInterceptEnable.1.1 true cTap2StreamStatus.1.1 createAndGo cTap2StreamType.1.1 = ip(1) cTap2StreamInterceptEnable.1.1 = true(1) cTap2StreamStatus.1.1 = createAndGo(4)`
	Example:
	To add TAP for L2VPN networks
	`setany -v3 <ip-address> <user> citapStreamInterface.4.1200 1125 citapStreamStatus.4.1200 createAndGo`	

Configuration Example for Inband Management Plane Feature Enablement

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

Configuring the Inband Management Plane Protection Feature: Example

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

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

Management Plane Protection - inband interface


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

Additional References

These sections provide references related to implementing lawful intercept.

Related Documents

Related Topic	Document Title
Lawful Intercept commands	*System Security Command Reference for Cisco ASR 9000 Series Routers*
Implementing SNMP	*System Management Configuration Guide for Cisco ASR 9000 Series Routers*
SNMP Server commands	*System Management Command Reference for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
A modular, open architecture designed for simple implementation that easily interacts with third-party equipment to meet service provider lawful intercept requirements.	See RFC-3924 under RFCs, on page 140.
An application layer protocol that facilitates the exchange of management information between network devices. Part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite.	Simple Network Management Protocol Version 3 (SNMPv3)

MIBs

MIBs	MIBs Link
• CISCO-TAP2-MIB, version 2 • CISCO-IP-TAP-MIB	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

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

Technical Assistance

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

CHAPTER 7

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 *System Security Command Reference for Cisco ASR 9000 Series Routers* .

Feature History for Implementing Management Plane Protection

Release	Modification
Release 3.7.2	This feature was introduced.

Prerequisites for Implementing Management Plane Protection

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

Restrictions for Implementing Management Plane Protection

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

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

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

Information About Implementing Management Plane Protection

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

Inband Management Interface

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

Out-of-Band Management Interface

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

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

Peer-Filtering on Interfaces

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

Control Plane Protection Overview

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

Management Plane

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

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

Management Plane Protection Feature

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

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

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

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

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

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

- SSH, v1 and v2
- SNMP, all versions

- Telnet
- TFTP
- HTTP
- HTTPS

Benefits of the Management Plane Protection Feature

Implementing the MPP feature provides the following benefits:

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

How to Configure a Device for Management Plane Protection

This section contains the following tasks:

Configuring a Device for Management Plane Protection for an Inband Interface

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

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

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

SUMMARY STEPS

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

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

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

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

Step 4	inband
	Example:
	`RP/0/RSP0/CPU0:router(config-mpp)# inband`
	`RP/0/RSP0/CPU0:router(config-mpp-inband)#`
		Configures an inband interface and enters management plane protection inband configuration mode.

Step 5	**interface** {*type instance* | **all**}
	Example:
	`RP/0/RSP0/CPU0:router(config-mpp-inband)# interface`
	` GigabitEthernet 0/6/0/1`
	`RP/0/RSP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)#`
		Configures a specific inband interface, or all inband interfaces. Use the **interface** command to enter management plane protection inband interface configuration mode.
		• Use the **all** keyword to configure all interfaces.

Step 6	**allow** {*protocol* | **all**} [**peer**]
	Example:
	`RP/0/RSP0/CPU0:router(config-mpp-inband-Gi0_6_0_1)#`
	` allow Telnet peer`
	`RP/0/RSP0/CPU0:router(config-telnet-peer)#`
		Configures an interface as an inband interface for a specified protocol or all protocols.
		• Use the *protocol* argument to allow management protocols on the designated management interface.
		  • HTTP or HTTPS
		  • SNMP (also versions)
		  • Secure Shell (v1 and v2)

	Command or Action	Purpose
		• TFTP • Telnet • Use the **all** keyword to configure the interface to allow all the management traffic that is specified in the list of protocols. • (Optional) Use the **peer** keyword to configure the peer address on the interface.
Step 7	**address ipv4** {*peer-ip-address* | *peer ip-address/length*} **Example:** `RP/0/RSP0/CPU0:router(config-telnet-peer)# address ipv4 10.1.0.0/16`	Configures the peer IPv4 address in which management traffic is allowed on the interface. • Use the *peer-ip-address* argument to configure the peer IPv4 address in which management traffic is allowed on the interface. • Use the *peer ip-address/length* argument to configure the prefix of the peer IPv4 address.
Step 8	**commit**	
Step 9	**show mgmt-plane** [**inband** | **out-of-band**] [**interface** {*type instance*}] **Example:** `RP/0/RSP0/CPU0:router# show mgmt-plane inband interface GigabitEthernet 0/6/0/1`	Displays information about the management plane, such as type of interface and protocols enabled on the interface. • (Optional) Use the **inband** keyword to display the inband management interface configurations that are the interfaces that process management packets as well as data-forwarding packets. • (Optional) Use the **out-of-band** keyword to display the out-of-band interface configurations. • (Optional) Use the **interface** keyword to display the details for a specific interface.

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

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

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

SUMMARY STEPS

1. **configure**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

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

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

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

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

Step 6	**interface** {*type instance* | **all**} **Example:**	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.

	Command or Action	Purpose
	`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)#`	• Use the **all** keyword to configure all interfaces.

Step 7	**allow** {*protocol* | **all**} [**peer**]
	Example:
	`RP/0/RSP0/CPU0:router(config-mpp-outband-Gi0_6_0_2)# allow TFTP peer`
	`RP/0/RSP0/CPU0:router(config-tftp-peer)#`
		Configures an interface as an out-of-band interface for a specified protocol or all protocols.
		• Use the *protocol* argument to allow management protocols on the designated management interface.
		  • HTTP or HTTPS
		  • SNMP (also versions)
		  • Secure Shell (v1 and v2)
		  • TFTP
		  • Telnet
		• Use the **all** keyword to configure the interface to allow all the management traffic that is specified in the list of protocols.
		• (Optional) Use the **peer** keyword to configure the peer address on the interface.

Step 8	**address ipv6** {*peer-ip-address* | *peer ip-address/length*} **Example:** `RP/0/RSP0/CPU0:router(config-tftp-peer)# address ipv6 33::33`	Configures the peer IPv6 address in which management traffic is allowed on the interface. • 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	**commit**	
Step 10	**show mgmt-plane** [**inband** | **out-of-band**] [**interface** {*type instance*} | **vrf**] **Example:** `RP/0/RSP0/CPU0:router# show mgmt-plane out-of-band interface GigabitEthernet 0/6/0/2`	Displays information about the management plane, such as type of interface and protocols enabled on the interface. • (Optional) Use the **inband** keyword to display the inband management interface configurations that are the interfaces that process management packets as well as data-forwarding packets. • (Optional) Use the **out-of-band** keyword to display the out-of-band interface configurations. • (Optional) Use the **interface** keyword to display the details for a specific interface.

	Command or Action	Purpose
		• (Optional) Use the **vrf** keyword to display the Virtual Private Network (VPN) routing and forwarding reference of an out-of-band interface.

Configuration Examples for Implementing Management Plane Protection

This section provides the following configuration example:

Configuring Management Plane Protection: Example

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

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

show mgmt-plane

Management Plane Protection

inband interfaces
----------------------
```

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

interface - all
        all configured -
                All peers allowed

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

show mgmt-plane out-of-band vrf

Management Plane Protection -
        out-of-band VRF - my_out_of_band
```

Additional References

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

Related Documents

Related Topic	Document Title
MPP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Management Plane Protection Commands on System Monitoring Command Reference for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 8

Configuring Software Authentication Manager

Software Authentication Manager (SAM) is a component of the 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 *System Security Command Reference for Cisco ASR 9000 Series Routers*.

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

Feature History for Configuring Software Authentication Manager

Release	Modification
Release 3.7.2	This feature was introduced.

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**sam promptinterval** *time-interval* {**proceed** | **terminate**} **Example:** `RP/0/RSP0/CPU0:router(config)# sam prompt-interval 25 {proceed | terminate}`	Sets the prompt interval in seconds, after which the SAM either proceeds or terminates the interval. The Prompt interval ranges from 0 to 300 seconds. If the user responds, SAM considers it as a ‘Yes’ and proceeds with the next action. If the user does not respond, SAM considers it as a ‘No’ and terminates the action. The default time for which SAM waits is 10 seconds.
Step 3	**commit**	

CHAPTER 9

Implementing Secure Shell

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

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

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

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

Note For a complete description of the Secure Shell commands used in this module , see the *Secure Shell Commands* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*.

Feature History for Implementing Secure Shell

Release	Modification
Release 3.7.2	This feature was introduced.
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 5.3.0	Support was added for Netconf Subsystem suppport on ssh server using a dedicated port. For more details see chapter *Implementing Network Configuration Protocol* in the *System Management Configuration Guide*.

Prerequisites for Implementing Secure Shell

The following prerequisites are required to implement Secure Shell:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Download the required image on your router. The SSH server and SSH client require you to have a a crypto package (data encryption standard [DES], 3DES and AES) from Cisco downloaded on your router.
- To run an SSHv2 server, you must have a VRF. This may be the default VRF or a specific VRF. VRF changes are applicable only to the SSH v2 server.
- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA). For more information, see the *Authentication, Authorization, and Accounting Commands on* Cisco IOS XR Software module in the *System Security Command Reference for Cisco ASR 9000 Series Routers* publication and *Configuring AAA Services on* Cisco IOS XR Software module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers* publication.
- AAA authentication and authorization must be configured correctly for Secure Shell File Transfer Protocol (SFTP) to work.

Restrictions for Implementing Secure Shell

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

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

- SFTP is compatible with sftp server version OpenSSH_2.9.9p2 or higher.
- RSA-based user authentication is supported in the SSH and SFTP servers. The support however, is not extended to the SSH client.
- Execution shell and SFTP are the only applications supported.
- The AES encryption algorithm is supported on the SSHv2 server and client, but not on the SSHv1 server and client. Any requests for an AES cipher sent by an SSHv2 client to an SSHv1 server are ignored, with the server using 3DES instead.
- The SFTP client does not support remote filenames containing wildcards (*, ?, []). The user must issue the **sftp** command multiple times or list all of the source files from the remote host to download them on to the router. For uploading, the router SFTP client can support multiple files specified using a wildcard provided that the issues mentioned in the first through third bullets in this section are resolved.
- The cipher preference for the SSH server follows the order AES128, AES192, AES256, and, finally, 3DES. The server rejects any requests by the client for an unsupported cipher, and the SSH session does not proceed.
- Use of a terminal type other than vt100 is unsupported, and the software generates a warning message in this case.
- Password messages of "none" are unsupported on the SSH client.
- Because the router infrastructure does not provide support for UNIX-like file permissions, files created on the local device lose the original permission information. For files created on the remote file system, the file permission adheres to the umask on the destination host and the modification and last access times are the time of the copy.

Information About Implementing Secure Shell

To implement SSH, you should understand the following concepts:

SSH Server

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

SSH Client

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

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

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

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

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

The SSH client supports the following options:

- DSCP—DSCP value for SSH client sessions.

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

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

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

- VRF—Source interface VRF for SSH client sessions:

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

SSH also supports remote command execution as follows:

```
RP/0/5/CPU0:router#ssh ?
  A.B.C.D  IPv4 (A.B.C.D) address
```

```
  WORD     Hostname of the remote node
  X:X::X   IPv6 (A:B:C:D...:D) address
  vrf      vrf table for the route lookup
RP/0/5/CPU0:router#ssh 1.1.1.1 ?
  cipher            Accept cipher type
  command           Specify remote command (non-interactive)
  source-interface  Specify source interface
  username          Accept userid for authentication
  <cr>
RP/0/5/CPU0:router#ssh 12.28.46.6 username admin command "show redundancy sum"
Password:


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

RP/0/5/CPU0:router#
```

SFTP Feature Overview

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

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

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

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

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

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

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

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

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

The SFTP server displays the following responses:

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

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

RSA Based Host Authentication

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

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

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

RSA Based User Authentication

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

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

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

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

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

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

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

Currently, only SSH version 2 and SFTP server support the RSA based authentication. For more information on how to import the public key to the router, see the *Implementing Certification Authority Interoperability on the Cisco ASR 9000 Series Router* chapter in this guide.

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

Authentication, Authorization, and Accounting (AAA) is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, see the *Authentication, Authorization, and Accounting Commands on* the Cisco ASR 9000 Series Router*Software* module in the *System Security Command Reference for Cisco ASR 9000 Series Routers* publication and the *Configuring AAA Services on* the Cisco ASR 9000 Series Router module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers* publication.

SSHv2 Client Keyboard-Interactive Authentication

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

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

For more information about the SSHv2 Client Keyboard-Interactive Authentication feature, see the *Implementing Secure Shell* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.For complete command reference of SSHv2 Client Keyboard-Interactive Authentication commands, see the *Secure Shell Commands*chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

How to Implement Secure Shell

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

Configuring SSH

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

SSH server supports setting DSCP value in the outgoing packets.

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

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

This is the syntax for setting DSCP value:

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

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

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

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

Perform this task to configure SSH.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**hostname** *hostname* **Example:** `RP/0/RSP0/CPU0:router(config)# hostname router1`	Configures a hostname for your router.
Step 3	**domain name** *domain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# domain name cisco.com`	Defines a default domain name that the software uses to complete unqualified host names.
Step 4	**commit**	
Step 5	**crypto key generate rsa** [**usage keys** | **general-keys**] [*keypair-label*] **Example:** `RP/0/RSP0/CPU0:router# crypto key generate rsa general-keys`	Generates an RSA key pair. The RSA key modulus can be in the range of 512 to 4096 bits. • To delete the RSA key pair, use the **crypto key zeroize rsa** command. • This command is used for SSHv1 only.
Step 6	**crypto key generate dsa** **Example:** `RP/0/RSP0/CPU0:router# crypto key generate dsa`	Enables the SSH server for local and remote authentication on the router. The supported key sizes are: 512, 768 and 1024 bits. • 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.
Step 7	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 8	**ssh timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# ssh timeout 60`	(Optional) Configures the timeout value for user authentication to AAA. • 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.

	Command or Action	Purpose

Step 9	Do one of the following:
	• **ssh server** [**vrf** *vrf-name* [**ipv4 access-list**IPv4 access-list name] [**ipv6 access-list** IPv6 access-list name]]
	• **ssh server v2**
	Example:
	`RP/0/RSP0/CPU0:router(config)# ssh`
	or
	`RP/0/RSP0/CPU0:router(config)# 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.Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened. To stop the SSH server from receiving any further connections for the specified VRF, use the **no** form of this command. If no VRF is specified, the default is assumed.
		Note 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	**commit**	
Step 11	**show ssh** **Example:** `RP/0/RSP0/CPU0:router# show ssh`	(Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router.
Step 12	**show ssh session details** **Example:** `RP/0/RSP0/CPU0:router# show ssh session details`	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Configuring the SSH Client

Perform this task to configure an SSH client.

SUMMARY STEPS

1. **configure**
2. **ssh client knownhost** *device* **:** /*filename*
3. **commit**
4. **ssh** {*ipv4-address* | *hostname*} [**username** *user- id* | **cipher des** | **source-interface** *type instance*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**ssh client knownhost** *device* **:** /*filename* **Example:** `RP/0/RSP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey`	(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end. **Note** The complete path of the filename is required. The colon (:) and slash mark (/) are also required.
Step 3	**commit**	
Step 4	**ssh** {*ipv4-address* | *hostname*} [**username** *user- id* | **cipher des** | **source-interface** *type instance*] **Example:** `RP/0/RSP0/CPU0:router# ssh remotehost username user1234`	Enables an outbound SSH connection. • To run an SSHv2 server, you must have a VRF. This may be the default or a specific VRF. VRF changes are applicable only to the SSH v2 server. • The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, the peer internally spawns an SSHv1 connection to the remote server. • The **cipher des** option can be used only with an SSHv1 client. • The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only. • If the *hostname* argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.

- If you are using SSHv1 and your SSH connection is being rejected, the reason could be that the RSA key pair might have been zeroed out or that you have not successfully generated an RSA key pair for your router. Another reason could be that the SSH server to which the user is connecting to using SSHv1 client does not accept SSHv1 connections. Make sure that you have specified a hostname and domain. Then use the **crypto key generate rsa** command to generate an RSA host-key pair, and then enable the SSH server.
- If you are using SSHv2 and your SSH connection is being rejected, the reason could be that the DSA, RSA host-key pair might have been zeroed out. Make sure you follow similar steps as mentioned above to generate the required host-key pairs, and then enable the SSH server.
- When configuring the RSA or DSA key pair, you might encounter the following error messages:
  - No hostname specified

  You must configure a hostname for the router using the **hostname** command.
  - No domain specified

  You must configure a host domain for the router using the **domain-name** command.

- The number of allowable SSH connections is limited to the maximum number of virtual terminal lines configured for the router. Each SSH connection uses a vty resource.
- From Cisco IOS XR Release 6.3.1 onwards, the **ssh client enable cipher** command is added for backward compatibility with the older Cisco IOS XR versions.

  For FIPS compliance, in Cisco IOS XR Releases later than 6.2.1, support for weaker ciphers like 3DES and AES CBC was removed and only AES-CTR cipher is supported.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.

Note If you are using Putty version 0.63 or higher to connect to the SSH client, set the 'Chokes on PuTTYs SSH2 winadj request' option under SSH > Bugs in your Putty configuration to 'On.' This helps avoid a possible breakdown of the session whenever some long output is sent from IOS XR to the Putty client.

Configuration Examples for Implementing Secure Shell

This section provides the following configuration example:

Configuring Secure Shell: Example

This example shows how to configure SSHv2 by creating a hostname, defining a domain name, enabling the SSH server for local and remote authentication on the router by generating a DSA key pair, bringing up the SSH server, and saving the configuration commands to the running configuration file.

After SSH has been configured, the SFTP feature is available on the router.

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate dsa
configure
ssh server
end
```

Multi-channeling in SSH

The multi-channeling (also called multiplexing) feature on the Cisco IOS XR software server allows you to establish multiple channels over the same TCP connection. Thus, rather than opening a new TCP socket for each SSH connection, all the SSH connections are multiplexed into one TCP connection. For example, with multiplexing support on your XR software server, on a single SSH connection you can simultaneously open a pseudo terminal, remotely execute a command and transfer a file using any file transfer protocol. Multiplexing offers the following benefits:

- You are required to authenticate only once at the time of creating the session. After that, all the SSH clients associated with a particular session use the same TCP socket to communicate to the server.
- Saves time consumed otherwise wasted in creating a new connection each time.

Multiplexing is enabled by default in the Cisco IOS XR software server. If your client supports multiplexing, you must explicitly set up multiplexing on the client for it to be able to send multi-channel requests to the server. You can use OpenSSH, Putty, Perl, WinSCP, Putty, FileZilla, TTSSH, Cygwin or any other SSH-based tool to set up multiplexing on the client. Configure Client for Multiplexing, on page 170 provides an example of how you can configure the client for multiplexing using OpenSSH.

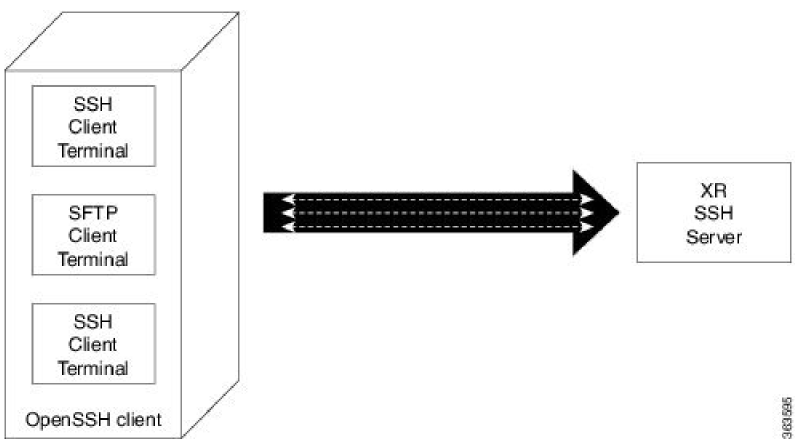
For more information on Multichannel feature, see the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 5.1.1.

Restrictions for Multi-channeling Over SSH

- Do not use client multiplexing for heavy transfer of data as the data transfer speed is limited by the TCP speed limit. Hence, for a heavy data transfer it is advised that you run multiple SSH sessions, as the TCP speed limit is per connection.
- Client multiplexing must not be used for more than 15 concurrent channels per session simultaneously.
- You must ensure that the first channel created at the time of establishing the session is always kept alive in order for other channels to remain open.

Client and Server Interaction Over Multichannel Connection

The figure below provides an illustration of a client-server interaction over a SSH multichannel connection.



As depicted in the illustration,

- The client multiplexes the collection of channels into a single connection. This allows different operations to be performed on different channels simultaneously. The dotted lines indicate the different channels that are open for a single session.
- After receiving a request from the client to open up a channel, the server processes the request. Each request to open up a channel represents the processing of a single service.

Note The Cisco IOX software supports server-side multiplexing only.

Configure Client for Multiplexing

The SSH client opens up one TCP socket for all the connections. In order to do so, the client multiplexes all the connections into one TCP connection. Authentication happens only once at the time of creating the session. After that, all the SSH clients associated with the particular session uses the same TCP socket to communicate to the server. Use the following steps to configure client multiplexing using OpenSSH:

SUMMARY STEPS

1. Edit the ssh_config file.
2. Add entries **ControlMaster auto** and **ControlPath**
3. Create a temporary folder.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Edit the ssh_config file.	Open the ssh_config file with your favorite text editor to configure values for session multiplexing. The system-wide SSH configuration file is located under /etc/ssh/ssh_config. The user configuration file is located under ~/.ssh/config or $HOME/.ssh/config.

Step 2	Add entries **ControlMaster auto** and **ControlPath**
	Example:
	`Host *`
	`ControlMaster auto`
	`ControlPath ~/.ssh/tmp/%r@%h:%p`
		Add the entry ControlMaster auto and ControlPath to the ssh_config file, save it and exit.
		• ControlMaster determines whether SSH will listen for control connections and what to do about them. Setting the ControlMaster to 'auto' creates a master session automatically but if there is a master session already available, subsequent sessions are automatically multiplexed.
		• ControlPath is the location for the control socket used by the multiplexed sessions. Specifying the ControlPath ensures that any time a connection to a particular server uses the same specified master connection.

Step 3	Create a temporary folder.	Create a temporary directory inside the /.ssh folder for storing the control sockets.

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

Related Topic	Document Title
AAA commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Authentication, Authorization, and Accounting Commands on* the Cisco ASR 9000 Series Router *Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*.
AAA configuration tasks	*Configuring AAA Services on* the Cisco ASR 9000 Series Router*Software* module in *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.
Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Host Services and Applications Commands on* the Cisco ASR 9000 Series Router module in *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.
IPSec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples **Note** IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).	*IPSec Network Security Commands on* the Cisco ASR 9000 Series Router*Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*
SSH commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Secure Shell Commands on* the Cisco ASR 9000 Series Router *Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
Draft-ietf-secsh-userauth-17.txt	*SSH Authentication Protocol*, July 2003
Draft-ietf-secsh-connect-17.txt	*SSH Connection Protocol*, July 2003
Draft-ietf-secsh-architecture-14.txt	*SSH Protocol Architecture*, July 2003
Draft-ietf-secsh-transport-16.txt	*SSH Transport Layer Protocol*, July 2003

MIBs

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

RFCs

RFCs	Title
RFC 6020	Netconf/ Yang

Technical Assistance

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

CHAPTER 10

Implementing Secure Socket Layer

This module describes how to implement SSL.

The Secure Socket Layer (SSL) protocol and Transport Layer Security (TLS) are application-level protocols that provide for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. SSL and TLS rely on certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Certificates are issued by certification authorities (CAs), such as VeriSign or Thawte. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date.

Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared quite freely, the private key is never given out. Each public-private key pair works together: Data encrypted with the public key can be decrypted only with the private key.

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

Feature History for Implementing Secure Socket Layer

Release	Modification
Release 3.7.2	This feature was introduced.

Prerequisites for Implementing Secure Shell

The following prerequisites are required to implement Secure Shell:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Download the required image on your router. The SSH server and SSH client require you to have a a crypto package (data encryption standard [DES], 3DES and AES) from Cisco downloaded on your router.
- To run an SSHv2 server, you must have a VRF. This may be the default VRF or a specific VRF. VRF changes are applicable only to the SSH v2 server.
- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA). For more information, see the *Authentication, Authorization, and Accounting Commands on* Cisco IOS XR Software module in the *System Security Command Reference for Cisco ASR 9000 Series Routers* publication and *Configuring AAA Services on* Cisco IOS XR Software module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers* publication.
- AAA authentication and authorization must be configured correctly for Secure Shell File Transfer Protocol (SFTP) to work.

Restrictions for Implementing Secure Shell

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

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

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

Information About Implementing Secure Shell

To implement SSH, you should understand the following concepts:

SSH Server

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

SSH Client

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

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

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

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

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

The SSH client supports the following options:

- DSCP—DSCP value for SSH client sessions.

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

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

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

- VRF—Source interface VRF for SSH client sessions:

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

SSH also supports remote command execution as follows:

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

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


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

RP/0/5/CPU0:router#
```

SFTP Feature Overview

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

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

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

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

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

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

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

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

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

The SFTP server displays the following responses:

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

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

RSA Based Host Authentication

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

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

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

RSA Based User Authentication

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

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

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

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

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

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

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

Currently, only SSH version 2 and SFTP server support the RSA based authentication. For more information on how to import the public key to the router, see the *Implementing Certification Authority Interoperability on the Cisco ASR 9000 Series Router* chapter in this guide.

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

Authentication, Authorization, and Accounting (AAA) is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, see the *Authentication, Authorization, and Accounting Commands on* the Cisco ASR 9000 Series Router*Software* module in the *System Security Command Reference for Cisco ASR 9000 Series Routers* publication and the *Configuring AAA Services on* the Cisco ASR 9000 Series Router module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers* publication.

SSHv2 Client Keyboard-Interactive Authentication

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

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

For more information about the SSHv2 Client Keyboard-Interactive Authentication feature, see the *Implementing Secure Shell* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.For complete command reference of SSHv2 Client Keyboard-Interactive Authentication commands, see the *Secure Shell Commands*chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

How to Implement Secure Shell

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

Configuring SSH

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

SSH server supports setting DSCP value in the outgoing packets.

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

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

This is the syntax for setting DSCP value:

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

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

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

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

Perform this task to configure SSH.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**hostname** *hostname* **Example:** `RP/0/RSP0/CPU0:router(config)# hostname router1`	Configures a hostname for your router.

	Command or Action	Purpose
Step 3	**domain name** *domain-name* **Example:** `RP/0/RSP0/CPU0:router(config)# domain name cisco.com`	Defines a default domain name that the software uses to complete unqualified host names.
Step 4	**commit**	
Step 5	**crypto key generate rsa** [**usage keys** | **general-keys**] [*keypair-label*] **Example:** `RP/0/RSP0/CPU0:router# crypto key generate rsa general-keys`	Generates an RSA key pair. The RSA key modulus can be in the range of 512 to 4096 bits. • To delete the RSA key pair, use the **crypto key zeroize rsa** command. • This command is used for SSHv1 only.
Step 6	**crypto key generate dsa** **Example:** `RP/0/RSP0/CPU0:router# crypto key generate dsa`	Enables the SSH server for local and remote authentication on the router. The supported key sizes are: 512, 768 and 1024 bits. • 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.
Step 7	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 8	**ssh timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# ssh timeout 60`	(Optional) Configures the timeout value for user authentication to AAA. • 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	Do one of the following: • **ssh server** [**vrf** *vrf-name* [**ipv4 access-list**IPv4 access-list name] [**ipv6 access-list** IPv6 access-list name]] • **ssh server v2** **Example:**	• (Optional) Brings up an SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed.Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened. To stop the SSH

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config)# ssh`
	or
	`RP/0/RSP0/CPU0:router(config)# ssh server v2`
		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.
		Note 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	**commit**	
Step 11	**show ssh** **Example:** `RP/0/RSP0/CPU0:router# show ssh`	(Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router.
Step 12	**show ssh session details** **Example:** `RP/0/RSP0/CPU0:router# show ssh session details`	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Configuring the SSH Client

Perform this task to configure an SSH client.

SUMMARY STEPS

1. **configure**
2. **ssh client knownhost** *device* **:** /*filename*
3. **commit**
4. **ssh** {*ipv4-address* | *hostname*} [**username** *user- id* | **cipher des** | **source-interface** *type instance*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**ssh client knownhost** *device* **:** /*filename* **Example:** `RP/0/RSP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey`	(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end. **Note** The complete path of the filename is required. The colon (:) and slash mark (/) are also required.
Step 3	**commit**	

	Command or Action	Purpose
Step 4	**ssh** {*ipv4-address* | *hostname*} [**username** *user- id* | **cipher des** | **source-interface** *type instance*] **Example:** `RP/0/RSP0/CPU0:router# ssh remotehost username user1234`	Enables an outbound SSH connection. • To run an SSHv2 server, you must have a VRF. This may be the default or a specific VRF. VRF changes are applicable only to the SSH v2 server. • The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, the peer internally spawns an SSHv1 connection to the remote server. • The **cipher des** option can be used only with an SSHv1 client. • The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only. • If the *hostname* argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.

- If you are using SSHv1 and your SSH connection is being rejected, the reason could be that the RSA key pair might have been zeroed out or that you have not successfully generated an RSA key pair for your router. Another reason could be that the SSH server to which the user is connecting to using SSHv1 client does not accept SSHv1 connections. Make sure that you have specified a hostname and domain. Then use the **crypto key generate rsa** command to generate an RSA host-key pair, and then enable the SSH server.
- If you are using SSHv2 and your SSH connection is being rejected, the reason could be that the DSA, RSA host-key pair might have been zeroed out. Make sure you follow similar steps as mentioned above to generate the required host-key pairs, and then enable the SSH server.
- When configuring the RSA or DSA key pair, you might encounter the following error messages:
    - No hostname specified

  You must configure a hostname for the router using the **hostname** command.
    - No domain specified

  You must configure a host domain for the router using the **domain-name** command.
- The number of allowable SSH connections is limited to the maximum number of virtual terminal lines configured for the router. Each SSH connection uses a vty resource.
- From Cisco IOS XR Release 6.3.1 onwards, the **ssh client enable cipher** command is added for backward compatibility with the older Cisco IOS XR versions.

  For FIPS compliance, in Cisco IOS XR Releases later than 6.2.1, support for weaker ciphers like 3DES and AES CBC was removed and only AES-CTR cipher is supported.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.

> **Note** If you are using Putty version 0.63 or higher to connect to the SSH client, set the 'Chokes on PuTTYs SSH2 winadj request' option under SSH > Bugs in your Putty configuration to 'On.' This helps avoid a possible breakdown of the session whenever some long output is sent from IOS XR to the Putty client.

Configuration Examples for Implementing Secure Shell

This section provides the following configuration example:

Configuring Secure Shell: Example

This example shows how to configure SSHv2 by creating a hostname, defining a domain name, enabling the SSH server for local and remote authentication on the router by generating a DSA key pair, bringing up the SSH server, and saving the configuration commands to the running configuration file.

After SSH has been configured, the SFTP feature is available on the router.

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate dsa
configure
ssh server
end
```

Multi-channeling in SSH

The multi-channeling (also called multiplexing) feature on the Cisco IOS XR software server allows you to establish multiple channels over the same TCP connection. Thus, rather than opening a new TCP socket for each SSH connection, all the SSH connections are multiplexed into one TCP connection. For example, with multiplexing support on your XR software server, on a single SSH connection you can simultaneously open a pseudo terminal, remotely execute a command and transfer a file using any file transfer protocol. Multiplexing offers the following benefits:

- You are required to authenticate only once at the time of creating the session. After that, all the SSH clients associated with a particular session use the same TCP socket to communicate to the server.
- Saves time consumed otherwise wasted in creating a new connection each time.

Multiplexing is enabled by default in the Cisco IOS XR software server. If your client supports multiplexing, you must explicitly set up multiplexing on the client for it to be able to send multi-channel requests to the server. You can use OpenSSH, Putty, Perl, WinSCP, Putty, FileZilla, TTSSH, Cygwin or any other SSH-based tool to set up multiplexing on the client. Configure Client for Multiplexing, on page 170 provides an example of how you can configure the client for multiplexing using OpenSSH.

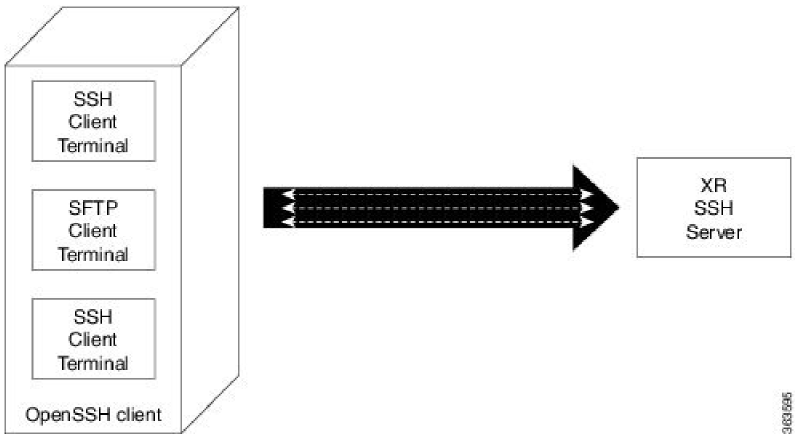
For more information on Multichannel feature, see the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 5.1.1.

Restrictions for Multi-channeling Over SSH

- Do not use client multiplexing for heavy transfer of data as the data transfer speed is limited by the TCP speed limit. Hence, for a heavy data transfer it is advised that you run multiple SSH sessions, as the TCP speed limit is per connection.
- Client multiplexing must not be used for more than 15 concurrent channels per session simultaneously.
- You must ensure that the first channel created at the time of establishing the session is always kept alive in order for other channels to remain open.

Client and Server Interaction Over Multichannel Connection

The figure below provides an illustration of a client-server interaction over a SSH multichannel connection.



As depicted in the illustration,

- The client multiplexes the collection of channels into a single connection. This allows different operations to be performed on different channels simultaneously. The dotted lines indicate the different channels that are open for a single session.
- After receiving a request from the client to open up a channel, the server processes the request. Each request to open up a channel represents the processing of a single service.

Note The Cisco IOX software supports server-side multiplexing only.

Configure Client for Multiplexing

The SSH client opens up one TCP socket for all the connections. In order to do so, the client multiplexes all the connections into one TCP connection. Authentication happens only once at the time of creating the session. After that, all the SSH clients associated with the particular session uses the same TCP socket to communicate to the server. Use the following steps to configure client multiplexing using OpenSSH:

SUMMARY STEPS

1. Edit the ssh_config file.
2. Add entries **ControlMaster auto** and **ControlPath**

3. Create a temporary folder.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Edit the ssh_config file.	Open the ssh_config file with your favorite text editor to configure values for session multiplexing. The system-wide SSH configuration file is located under /etc/ssh/ssh_config. The user configuration file is located under ~/.ssh/config or $HOME/.ssh/config.

Step 2	Add entries **ControlMaster auto** and **ControlPath**
	Example:
	`Host *`
	`ControlMaster auto`
	`ControlPath ~/.ssh/tmp/%r@%h:%p`
		Add the entry ControlMaster auto and ControlPath to the ssh_config file, save it and exit.
		• ControlMaster determines whether SSH will listen for control connections and what to do about them. Setting the ControlMaster to 'auto' creates a master session automatically but if there is a master session already available, subsequent sessions are automatically multiplexed.
		• ControlPath is the location for the control socket used by the multiplexed sessions. Specifying the ControlPath ensures that any time a connection to a particular server uses the same specified master connection.

Step 3	Create a temporary folder.	Create a temporary directory inside the /.ssh folder for storing the control sockets.

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

Related Topic	Document Title
AAA commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Authentication, Authorization, and Accounting Commands on* the Cisco ASR 9000 Series Router *Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*.
AAA configuration tasks	*Configuring AAA Services on* the Cisco ASR 9000 Series Router*Software* module in *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.
Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Host Services and Applications Commands on* the Cisco ASR 9000 Series Router module in *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.

Related Topic	Document Title
IPSec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples **Note** IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).	*IPSec Network Security Commands on* the Cisco ASR 9000 Series Router*Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*
SSH commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Secure Shell Commands on* the Cisco ASR 9000 Series Router *Software* module in *System Security Command Reference for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
Draft-ietf-secsh-userauth-17.txt	*SSH Authentication Protocol*, July 2003
Draft-ietf-secsh-connect-17.txt	*SSH Connection Protocol*, July 2003
Draft-ietf-secsh-architecture-14.txt	*SSH Protocol Architecture*, July 2003
Draft-ietf-secsh-transport-16.txt	*SSH Transport Layer Protocol*, July 2003

MIBs

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

RFCs

RFCs	Title
RFC 6020	Netconf/ Yang

Technical Assistance

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

CHAPTER 11

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

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

Table 10: Security Features for Layer 2 VPNs

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

CHAPTER 12

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.

Traffic storm control can be configured at the bridge domain level. Support has been added to allow storm control rate to be configured in kbps. For more information about the Traffic Storm Control feature, see the *Implementing Traffic Storm Control under a VPLS Bridge* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of Traffic Storm Control commands, see the *Traffic Storm Control Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*.

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 type for the remainder of the 1-second interval. At the beginning of a new 1-second interval, traffic of the specified type is allowed to pass on the port.

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.

> **Note**
> - Bridge Protocol Data Unit (BPDU) packets are not filtered through the storm control feature.
> - Tunneled BPDU packets are filtered as they are forwarded into bridge.
> - Traffic storm control is applied to only forwarded packets in the system.

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.

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. **interface** *interface-name*
6. **storm-control** {**broadcast** | **multicast** | **unknown-unicast**} **pps** *packet-threshold*
7. **commit**
8. **show l2vpn bridge-domain bd-name** *bridge-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	l2vpn
	Example:
	`RP/0/0/CPU0:router(config)# l2vpn`
	`RP/0/0/CPU0:router(config-l2vpn)#`
		Enters L2 VPN configuration mode.

Step 3	**bridge group** *bridge-group-name* **Example:**	Enters L2 VPN bridge group configuration mode.

	Command or Action	Purpose
	`RP/0/0/CPU0:router(config-l2vpn)# bridge group csco` `RP/0/0/CPU0:router(config-l2vpn-bg)#`	

Step 4	**bridge-domain** *bridge-domain-name*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg)# bridge-domain abc`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd)#`
		Enters L2 VPN bridge domain configuration mode.

Step 5	**interface** *interface-name*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/1/0/0.100`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-ac)#`
		Names an AC under the bridge domain. In this case, the AC is a VLAN on an Ethernet interface.

Step 6	**storm-control** {**broadcast** | **multicast** | **unknown-unicast**} **pps** *packet-threshold*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-ac)# storm-control broadcast pps 4500`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-ac)# storm-control multicast pps 500`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-ac)#`
		Enables traffic storm control on this interface for the specified traffic type. Repeat this command for each traffic type.
		The *packet-threshold* is a packet per second rate and must be an integer between 1 and 160000. It specifies the number of packets that will be allowed to pass on the interface for the specified traffic type during a 1-second interval.

Step 7	**commit**	
Step 8	**show l2vpn bridge-domain bd-name** *bridge-name* **detail** **Example:** `RP/0/0/CPU0:router# show l2vpn bridge-domain bd-name abc detail`	Displays storm control configuration.

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. **commit**
8. **show l2vpn bridge-domain bd-name** *bridge-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	l2vpn
	Example:
	`RP/0/0/CPU0:router(config)# l2vpn`
	`RP/0/0/CPU0:router(config-l2vpn)#`
		Enters L2 VPN configuration mode.

Step 3	**bridge group** *bridge-group-name*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn)# bridge group csco`
	`RP/0/0/CPU0:router(config-l2vpn-bg)#`
		Enters L2 VPN bridge group configuration mode.

Step 4	**bridge-domain** *bridge-domain-name*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg)# bridge-domain abc`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd)#`
		Enters L2 VPN bridge domain configuration mode.

Step 5	**neighbor** *address* **pw-id** *id*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd)# neighbor 1.1.1.1 pw-id 100`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)#`
		Names an access pseudowire under the bridge domain.
		Note You cannot apply storm control on a forwarding PW (a PW under a VFI).

Step 6	**storm-control** {**broadcast** | **multicast** | **unknown-unicast**} **pps** *packet-threshold*
	Example:
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)# storm-control broadcast pps 4500`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)# storm-control multicast pps 500`
	`RP/0/0/CPU0:router(config-l2vpn-bg-bd-pw)#`
		Enables traffic storm control on this pseudowire for the specified traffic type. Repeat this command for each traffic type.
		The *packet-threshold* is a packet per second rate and must be an integer between 1 and 160000. It specifies the number of packets that will be allowed to pass on the interface for the specified traffic type during a 1-second interval.

Step 7	**commit**	

	Command or Action	Purpose
Step 8	**show l2vpn bridge-domain bd-name** *bridge-name* **detail** **Example:** `RP/0/0/CPU0:router# show l2vpn bridge-domain bd-name csco detail`	Displays storm control configuration settings for the named bridge domain. This command also displays the drop counter values for each configured storm control instance.

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

	Command or Action	Purpose
Step 1	clear l2vpn forwarding counters **Example:** `RP/0/0/CPU0:router# clear l2vpn forwarding counters`	Clears l2vpn forwarding counters, including storm control drop counters.

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
   interface GigabitEthernet0/1/0/3.215
    storm-control multicast pps 500
    storm-control broadcast pps 4500
   !
[lines deleted]


RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 215 detail
Bridge group: 215, bridge-domain: 215, id: 3, 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: 9000
  Filter MAC addresses:
  ACs: 2 (2 up), VFIs: 1, PWs: 1 (1 up)
  List of ACs:
    AC: GigabitEthernet0/1/0/3.215, state is up
      Type VLAN; Num Ranges: 1
     vlan ranges: [100, 100]
      MTU 9008; XC ID 0x440005; interworking none; MSTi 0 (unprotected)
      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:
       Broadcast: enabled(4500)
       Multicast: enabled(500)
       Unknown unicast: disabled
      Static MAC addresses:
      Statistics:
        packet totals: receive 36728, send 31
        byte totals: receive 2791284, send 2318
        Storm control drop counters:
          packet totals: broadcast 0, multicast 0, unknown unicast 0
          byte totals: broadcast 0, multicast 0, unknown unicast 0
[lines deleted]
```

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
l2vpn
 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)
Last time status changed: 16/12/2008 00:35:02 (00:31:28 ago)
  MAC withdraw message: send 0 receive 0
Static MAC addresses:
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
Storm control drop counters:
  packets: broadcast 0, multicast 0, unknown unicast 0
  bytes: broadcast 0, multicast 0, unknown unicast 0
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:
 Broadcast: enabled(100)
 Multicast: enabled(110)
 Unknown unicast: enabled(120)
```

Additional References

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

Related Documents

Related Topic	Document Title
MPLS Layer 2 VPNs	*Implementing MPLS Layer 2 VPNs on Cisco ASR 9000 Series Router* module in the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*.
MPLS VPLS bridges	*Implementing Virtual Private LAN Services on Cisco ASR 9000 Series Router* module in the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*
Getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*

Standards

Standards [1]	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

[1] Not all supported standards are listed.

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 13

Configuring FIPS Mode

The Federal Information Processing Standard (FIPS) 140-2 is an U.S. and Canadian government certification standard that defines requirements that the cryptographic modules must follow. The FIPS specifies best practices for implementing cryptographic algorithms, handling key material and data buffers, and working with the operating system.

In Cisco IOS XR software, these applications are verified for FIPS compliance:

- Secure Shell (SSH)
- Secure Socket Layer (SSL)
- Transport Layer Security (TLS)
- Internet Protocol Security (IPSec) for Open Shortest Path First version 3 (OSPFv3)
- Simple Network Management Protocol version 3 (SNMPv3)

For more information about the Configuring FIPS feature, see the *Configuring FIPS Mode* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of FIPS commands, see the *FIPS Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

Prerequisites for Configuring FIPS

Install and activate the **asr9k-k9sec-px.pie** file.

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.

Installing and Activating the PIE

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

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

You must install and activate the **asr9k-k9sec-px.pie** file to configure FIPS. To install and activate the PIE, download the **asr9k-k9sec-px.pie** to a TFTP server.

For more information about installing PIEs, refer to *Upgrading and Managing Cisco IOS XR Software section* of the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**install add** *tftp://<IP address of tftp server>/<location of pie on server>* **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://172.201.11.140/auto/tftp-users1/pie/`	Copies the contents of a package installation envelope (PIE) file to a storage device.
Step 3	**install activate** *device:package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-k9sec-px.pie`	Activates the respective package and adds more functionality to the existing software.
Step 4	**install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Saves the active software set to be persistent across designated system controller (DSC) reloads.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits from the admin mode.

	Command or Action	Purpose
Step 6	**show install committed** **Example:** `RP/0/RSP0/CPU0:router# show install committed`	Shows the list of the committed software packages.

How to Configure FIPS

Perform these tasks to configure FIPS.

Enabling FIPS mode

Before you begin

Refer to the Installing and Activating the PIE, on page 204 section for information on installing and activating the image on the router.

SUMMARY STEPS

1. **configure**
2. **crypto fips-mode**
3. **commit**
4. **show logging**
5. **admin**
6. **reload location all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**crypto fips-mode** **Example:** `RP/0/RSP0/CPU0:router(config)#crypto fips-mode`	Enters FIPS configuration mode. **Note** Stop new incoming SSH sessions while configuring or unconfiguring **crypto fips-mode**. Restart the router upon configuration.
Step 3	**commit**	
Step 4	**show logging** **Example:** `RP/0/RSP0/CPU0:router#show logging`	Displays the contents of logging buffers. **Note** Use the **show logging | i fips** command to filter FIPS specific logging messages.
Step 5	**admin** **Example:** `RP/0/RSP0/CPU0:router#admin`	Enters into the admin EXEC mode.

	Command or Action	Purpose
Step 6	**reload location all** **Example:** `RP/0/RSP0/CPU0:router(admin)#reload location all`	Reloads a node or all nodes on a single chassis or multishelf system.

Configuring FIPS-compliant Keys

Perform these steps to configure the FIPS-compliant keys:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa [usage-keys | general-keys]** *key label* **Example:** `RP/0/RSP0/CPU0:router#crypto key generate rsa general-keys rsakeypair`	Generate a RSA key pair. Ensure that all the key pairs meet the FIPS requirements. The length of the key can vary from 1024 to 2048 bits. The option **usage-keys** generates separate RSA key pairs for signing and encryption. The option **general-keys** generates a general-purpose RSA key pair for signing and encryption. To delete the RSA key pair, use the **crypto key zeroize rsa** *keypair-label* command.
Step 2	**crypto key generate dsa** **Example:** `RP/0/RSP0/CPU0:router#crypto key generate dsa`	Generate a DSA key pair if required. Ensure that all the key pairs meet the FIPS requirements. The length of the key is restricted to1024 bits. To delete the DSA key pair, use the **crypto key zeroize dsa** *keypair-label* command.
Step 3	**show crypto key mypubkey rsa** **Example:** `RP/0/RSP0/CPU0:router#show crypto key mypubkey rsa`	Displays the existing RSA key pairs
Step 4	**show crypto key mypubkey dsa** **Example:**	Displays the existing DSA key pairs

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router#show crypto key mypubkey dsa`	

Configuring FIPS-compliant Key Chain

Perform these steps to configure the FIPS-compliant key chain:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

1. **configure**
2. **key chain** *key-chain-name*
3. **key** *key-id*
4. **cryptographic-algorithm** {**HMAC-SHA1-20** | **SHA-1**}
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router#configure`	Enters the global configuration mode.
Step 2	**key chain** *key-chain-name* **Example:** `RP/0/RSP0/CPU0:router(config)#key chain mykeychain`	Creates a key chain.
Step 3	**key** *key-id* **Example:** `RP/0/RSP0/CPU0:router(config-mykeychain)#key 1`	Creates a key in the key chain.
Step 4	**cryptographic-algorithm** {**HMAC-SHA1-20** | **SHA-1**} **Example:** `RP/0/RSP0/CPU0:router(config-mykeychain-1)#cryptographic-algorithm HMAC-SHA1-20`	Configures the cryptographic algorithm for the key chain. Ensure that the key chain configuration always uses SHA-1 as the hash or keyed hash message authentication code (hmac) algorithm.
Step 5	**commit**	

Configuring FIPS-compliant Certificates

Perform these steps to configure the FIPS-compliant certificates:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

1. **configure**
2. **crypto ca trustpoint** *ca-name key label*
3. **commit**
4. **show crypto ca certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**crypto ca trustpoint** *ca-name key label* **Example:** `RP/0/RSP0/CPU0:router(config)#crypto ca trustpoint msiox rsakeypair`	Configures the trustpoint by utilizing the desired RSA keys. Ensure that the certificates meet the FIPS requirements for key length and signature hash or encryption type. **Note** The minimum key length for RSA or DSA key is 1024 bits. The required hash algorithm is SHA-1-20.
Step 3	**commit**	
Step 4	**show crypto ca certificates** **Example:** `RP/0/RSP0/CPU0:router#show crypto ca certificates`	Displays the information about the certificate

Configuring FIPS-compliant OSPFv3

Perform these steps to configure the FIPS-compliant OSPFv3:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

1. **configure**
2. **router ospfv3** *process name*
3. **area** *id*
4. **authentication**{**disable** | **ipsec spi** *spi-value* **sha1** [**clear** | **password**] *password*}
5. **exit**
6. **encryption**{**disable** | {**ipsec spi** *spi-value* **esp** {**3des** | **aes** [**192** | **256**] [**clear** |**password**] *encrypt-password*} [**authentication** **sha1**[**clear** | **password**] *auth-password*]}}
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router ospfv3** *process name* **Example:** `RP/0/RSP0/CPU0:router(config)#router ospfv3 ospfname`	Configures the OSPFv3 process.
Step 3	**area** *id* **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3)#area 1`	Configures the OSPFv3 area ID. The ID can either be a decimal value or an IP address.
Step 4	**authentication** { **disable** | **ipsec spi** *spi-value* **sha1** [**clear** | **password**] *password* } **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3-ar)#authentication ipsec spi 256 sha1 password pa1`	Enables authentication for OSPFv3. Note that the OSPFv3 configuration supports only SHA-1 for authentication. **Note** IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3-ar)#exit`	Exits OSPFv3 area configuration and enters the OSPFv3 configuration mode.
Step 6	**encryption** { **disable** | { **ipsec spi** *spi-value* **esp** { **3des** | **aes** [**192** | **256**] [**clear** | **password**] *encrypt-password* } [**authentication sha1** [**clear** | **password**] *auth-password*] } } **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3)#encryption ipsec spi 256 esp 3des password pwd`	Encrypts and authenticates the OSPFv3 packets. Ensure that the OSPFv3 configuration uses the following for encryption in the configuration. • 3DES: Specifies the triple DES algorithm. • AES: Specifies the Advanced Encryption Standard (AES) algorithm. Ensure that SHA1 is chosen if the authentication option is specified.
Step 7	**commit**	

Configuring FIPS-compliant SNMPv3 Server

Perform these steps to configure the FIPS-compliant SNMPv3 server:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

1. **configure**

2. **snmp-server user** *username groupname* {**v3** [**auth sha** {**clear** | **encrypted**} *auth-password* [**priv** {**3des** |**aes** { **128** | **192** | **256**} } {**clear** | **encrypted** } *priv-password*]] } [**SDROwner** | **SystemOwner**] *access-list-name*
3. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router#configure`	Enters the global configuration mode.
Step 2	**snmp-server user** *username groupname* {**v3** [**auth sha** {**clear** | **encrypted**} *auth-password* [**priv** {**3des** |**aes** { **128** | **192** | **256**} } {**clear** | **encrypted** } *priv-password*]] } [**SDROwner** | **SystemOwner**] *access-list-name* **Example:** `RP/0/RSP0/CPU0:router(config)#snmp-server user user1 g v3 auth sha clear pass priv aes 128 clear privp`	Configures the SNMPv3 server.
Step 3	**commit**	

Configuring FIPS-compliant SSH Client and Server

Perform these steps to configure the FIPS-compliant SSH Client and the Server:

Before you begin

Refer the configuration steps in the Enabling FIPS mode, on page 205 section for enabling the FIPS mode.

SUMMARY STEPS

1. **ssh** {*ipv4-address* | *ipv6-address*} **cipher aes** {**128-CTR** | **192-CTR** | **256-CTR**} **username** *username*
2. **configure**
3. **ssh server v2**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ssh** {*ipv4-address* | *ipv6-address*} **cipher aes** {**128-CTR** | **192-CTR** | **256-CTR**} **username** *username* **Example:**	Configures the SSH client. Ensure that SSH client is configured only with the FIPS-approved ciphers. AES(Advanced Encryption Standard)-CTR (Counter mode) is the FIPS-compliant cipher algorithm with key lengths of 128, 192 and 256 bits.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router#ssh 10.1.2.3 cipher aes 128-CTR username user1`	
Step 2	**configure**	
Step 3	**ssh server v2** **Example:** `RP/0/RSP0/CPU0:router(config)#ssh server v2`	Configures the SSH server. The following dh algorithms are supported on the SSH client: **1.** ecdh-sha2-nistp256 **2.** ecdh-sha2-nistp384 **3.** ecdh-sha2-nistp521
Step 4	**commit**	

Configuration Examples for Configuring FIPS

This section provides examples for configuring FIPS.

Configuring FIPS: Example

This example shows how to configure FIPS:

```
RP/0/3/CPU0:SSH#configure
RP/0/3/CPU0:SSH(config)#crypto fips-mode
RP/0/3/CPU0:SSH(config)#commit
RP/0/3/CPU0:SSH(config)#end
```

This example shows the output of **show logging** command:

```
RP/0/3/CPU0:SSH(config)#crypto fips-mode
RP/0/3/CPU0:SSH(config)#commit
RP/0/3/CPU0:SSH(config)#end
RP/0/3/CPU0:SSH#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 60 messages logged
    Monitor logging: level debugging, 0 messages logged
    Trap logging: level informational, 0 messages logged
    Buffer logging: level debugging, 3 messages logged

Log Buffer (9000000 bytes):
<output omitted>

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 16 12:48:17.736 : cepki[433]: The configuration setting for FIPS mode
has been modified. The system must be reloaded to finalize this configuration change. Please
 refer to the IOS XR System Security Configuration Guide, Federal Information Process
Standard(FIPS) Overview section when modifying the FIPS mode setting.
RP/0/RSP0/CPU0:Apr 16 12:48:17.951 : config[65757]: %MGBL-CONFIG-6-DB_COMMIT :
```

```
Configuration committed by user 'lab'. Use 'show configuration commit changes 1000000002'
to view the changes.
RP/0/RSP0/CPU0:Apr 16 12:48:23.988 : config[65757]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by lab


....
....
....
```

CHAPTER 14

Implementing Cisco ASR 9000 vDDoS Mitigation

This module provides information about how to implement Cisco ASR 9000 vDDoS mitigation to protect network infrastructures and resources from distributed denial-of-service (DDoS) attacks.

Cisco ASR 9000 vDDoS Mitigation Overview

Distributed denial-of-service (DDoS) attacks target network infrastructures or computer services resources. The primary goal of DDoS attacks is to deny legitimate users access to a particular computer or network resources, which results in service degradation, loss of reputation, and irretrievable data loss. DDoS Mitigation is the process of detecting increasingly complex and deceptive assaults and mitigating the effects of the attack to ensure business continuity and resource availability.

The Arbor Peakflow solution protects customer networks by mitigating undesirable traffic caused by DDoS attacks. It comprises a number of functions as well as a set of hardware devices that implement those functions. Peakflow SP means the control components such as monitoring the network, detecting attacks, and coordinating an attack response. Peakflow SP runs on SP appliances or in virtual machines. Peakflow Threat Management System (TMS) or Peakflow SP TMS is the data plane component to remove DDoS attacks.

Using Netflow and BGP, Arbour Peakflow solution monitors the network ingress points to build a base line for network behavior and traffic patterns. It will then perform ongoing monitoring to detect anomalies and flag them as potential attacks. These potential attacks are presented to network operations via a GUI, email, or SNMP which allows a range of actions to be taken, including initiating a response or marking an event as a false alarm. If there is an attack, the Arbor Peakflow solution redirects all traffic for the destination through the TMS which can remove unwanted traffic and clean the traffic as effectively as possible without blocking valid connections. The new path to the TMS where the traffic from the original path is diverted is called off-ramp traffic path. The path from the TMS egress interface to the original destination of the traffic where the clean traffic is sent is called on-ramp traffic path.

Cisco has partnered with Arbor Networks to deliver DDoS attack mitigation capabilities on Cisco ASR 9000 Series routers by integrating the Threat Management System (TMS) DDoS mitigation functionality to the Cisco ASR 9000 router. The TMS will be implemented on the ASR 9000 VSM (Virtualized Services Module) hosted in the ASR 9000 chassis.

Information about Implementing Cisco ASR 9000 vDDoS Mitigation

There are different ways to implement DDoS mitigation. In the centralized model, a dedicated part of the network will be the scrubbing center (TMS) to clean the traffic and the traffic to the victim will be diverted to the scrubbing center. In the distributed approach, scrubbers are installed at the edge of the network. In the mixed approach, scrubbers will be present at the edge and the scrubbing center will handle the additional traffic. You should choose the mitigation strategy suitable for your network.

The mechanisms to create an effective diversion and re-injection path include BGP Flowspec, injecting a more specific route by diverting traffic to the victim in to the TMS, tunneling traffic to the TMS and from the TMS, putting the malicious and clean traffic in different VRFs or VPNs, and using ACL Based Forwarding (ABF) to steer traffic. These tools can be used in different combinations like tunnel diversion & VRF re-injection, diversion using a /32 prefix and VPN re-injection, and /32 diversion and GRE tunnel re-injection to implement a range of routing designs.

Prerequisites for Implementing Cisco ASR 9000 vDDoS Mitigation

These prerequisites are required to implement DDoS Mitigation support on the Cisco ASR 9000 Series Router.

- You need Cisco IOS XR software release 5.3.0 or later installed on the Cisco ASR 9000 Series Router.
- ASR 9000 Series Route Switch Processor 440 (RSP 440) or above is required.
- You need to insert the VSM card in the Cisco ASR 9000 Series Router.
- TFTP should be enabled on the CIsco ASR 9000 Series Router.
- You need to uninstall any pre-existing virtual service on the VSM card.
- You need to pair the ASR 9000 vDDoS solution with Arbor Peakflow SP.

Restrictions for Implementing Cisco ASR 9000 vDDoS Mitigation

The following restriction apply for implementing Cisco ASR 9000 vDDoS mitigation.

- Only one vDDoS instance is supported per VSM card.

Configuring Cisco ASR 9000 vDDoS Mitigation

This section provides information about the configuration tasks required for implementing ASR 9000 vDDoS mitigation. This section only provides information about Cisco ASR9000 specific configuration. For Arbor Peakflow SP configuration, see *Arbor Networks SP and Threat Management System (TMS) User Guide* .

Installing Cisco ASR 9000 vDDoS Software

Arbor Networks TMS and ArbOS are packaged together with configuration files in an Open Virtualization Archive (.ova) file. Installation of ASR 9000 vDDoS software on the VSM card consists of the following steps:

1. Copy the OVA file that contains Arbor TMS and Arbor OS to the ASR 9000 router using TFTP or FTP. Use the correct path and filename for your build. When you are prompted for the remote host, type the IP address of the remote host. For destination file name, press enter.

```
RP/0/RSP0/CPU0:router# copy tftp:/Peakflow-TMS-8.0.0-EKU0.ova disk0:
```

2. Enable the virtual service.

```
RP/0/RSP0/CPU0:router# virtual-service enable
RP/0/RSP0/CPU0:router# commit
```

3. Install the TMS VSM software.

```
RP/0/RSP0/CPU0:router# virtual-service install name tms3 package
/disk0:/Peakflow-TMS-8.0.0-EKU0.ova node 0/1/CPU0
```

The installation may take 10-12 minutes to complete.

4. Check the progress of the installation process by using the **show virtual-service list** command.

```
RP/0/RSP0/CPU0:router# show virtual-service list
```

If installation is in process, this command shows the status as installing. When installation is complete, you can rerun this show command to verify that the virtual service is listed as installed.

Configuring Interfaces for TMS Mitigation

Once you install the VSM module, twelve virtual Network Interface Card (vNIC) interfaces are available between the VSM module and the router. You can use some of these vNIC interfaces for TMS mitigation and others for management of the TMS virtual instance. The mitigation interfaces are bundled into a single logical interface. The logical interface can be divided into subinterfaces for diversion and re-injection of traffic.

1. Map vNIC interfaces on the router to TMS interfaces on the VSM card.

```
RP/0/RSP0/CPU0:router(config)# virtual-service tms3
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/0
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/1
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/2
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/3
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/4
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/5
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/6
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/7
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/8
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/9
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/10
RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/1/1/11
RP/0/RSP0/CPU0:router(config-virt-service)# commit
RP/0/RSP0/CPU0:router(config-virt-service)# activate
RP/0/RSP0/CPU0:router(config-virt-service)# commit
```

2. Check the progress of the activation process by using the **show virtual-service list** command.

```
RP/0/RSP0/CPU0:router# show virtual-service list
```

Once the VM is activated, the status changes to activated.

3. Create ethernet bundle interface for mitigation interfaces 0-3 and 7-10.

```
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether  2
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
RP/0/RSP0/CPU0:router(config-if)# bundle load-balancing hash src-ip
RP/0/RSP0/CPU0:router(config-if)# exit
```

4. Add member interfaces to the ethernet bundle.

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/0
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/1
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/2
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/3
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit


RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/7
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/8
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/9
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/10
RP/0/RSP0/CPU0:router(config-if)# bundle-id 2 mode on
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit
```

5. Configure TMS management interfaces 5 and 6.

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/5
RP/0/RSP0/CPU0:router(config-if)# ip address 10.2.1.10 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/6
RP/0/RSP0/CPU0:router(config-if)# ip address 10.2.1.5 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)# load interval 30
RP/0/RSP0/CPU0:router(config-if)# commit
```

6. Set up unused interfaces 4 and 11.

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/4
RP/0/RSP0/CPU0:router(config-if)# shut down
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/1/11
RP/0/RSP0/CPU0:router(config-if)# shut down
RP/0/RSP0/CPU0:router(config-if)# commit
```

7. Configure subinterfaces for diversion and re-injection.

```
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 2.100
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.1.2.100 255.255.255.240
RP/0/RSP0/CPU0:router(config-subif)#  bundle load-balancing hash src-ip
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 2.101
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RSP0/CPU0:router(config-subif)# vrf onramp
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.1.2.130 255.255.255.240
```

Uninstalling the TMS Virtual Service

Before installing the TMS software on VSM card, you need to remove any existing TMS virtual service on the VSM card. Perform the followings steps to remove any instances of the TMS virtual service.

1. Enable the virtual services on the VSM card.

```
RP/0/RSP0/CPU0:router(config)# virtual-service enable
RP/0/RSP0/CPU0:router(config)# commit
```

2. Use the **show virtual-service list** command to see the list of virtual services available on the VSM card.

```
RP/0/RSP0/CPU0:router# show virtual-service list
```

3. If the TMS virtual instance is listed, de-activate the TMS virtual instance.

```
RP/0/RSP0/CPU0:router(config)# no virtual-service tms3
RP/0/RSP0/CPU0:router(config)# commit
```

4. Uninstall the TMS virtual instance.

```
RP/0/RSP0/CPU0:router# virtual-service uninstall name tms3 node 0/1/CPU0
```