



Configuring AAA Services on Cisco IOS XR Software

This module describes the implementation of the administrative model of *task-based authorization* used to control user access in the Cisco IOS XR 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 on Cisco IOS XR Software* module in the *Cisco IOS XR System Security Command Reference* publication. 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 on Cisco CRS-1

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	No modification.

Release 3.3.0	<ul style="list-style-type: none"> • Support for the RADIUS Dead-Server Detection feature was added. • To enable interoperability based on Cisco IOS software, tasks must be marked as an optional attribute. • Support was added on Cisco IOS XR to allow 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 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. • A procedure on how to specify the task ID and user groups by using the CiscoSecure ACS was added. • All references to owner secure domain router (SDR) were replaced with root SDR. • Support was added to prompt the next logged-in user for a new username and password if all the users were deleted. • The predefined task group <i>serviceadmin</i> was added. • An example was added for RADIUS Vendor -Specific Attribute (VSA). • EXEC authorization was added to the “Administrative Access” section.
Release 3.4.0	<ul style="list-style-type: none"> • The server-private command was added to configure RADIUS server groups. • Support for the Per VRF AAA feature was added. • Support for generating interim accounting records was added.
Release 3.5.0	No modification.
Release 3.6.0	The eventmanager keyword (fault manager) replaces the exec keyword to authorize event managers (fault managers) from the aaa authorization command.
Release 3.7.0	No modification.
Release 3.8.0	Information was edited to make clearer which features are supported on the Cisco CRS-1 exclusively.
Release 3.9.0	No modification.

Contents

- [Prerequisites for Configuring AAA Services, page SC-3](#)
- [Restrictions for Configuring AAA Services, page SC-3](#)
- [Information About Configuring AAA Services, page SC-3](#)
- [How to Configure AAA Services, page SC-18](#)
- [Configuration Examples for Configuring AAA Services, page SC-56](#)
- [Additional References, page SC-58](#)

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](#)” 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](#), page SC-4
- [Cisco IOS XR Software Administrative Model](#), page SC-6
- [Password Types](#), page SC-11
- [Task-Based Authorization](#), page SC-11
- [Task IDs for TACACS+ and RADIUS Authenticated Users](#), page SC-14
- [XML Schema for AAA Services](#), page SC-16
- [About RADIUS](#), page SC-17

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](#)” section)

User Categories

Router users are classified into the following categories:

- Root system user (complete administrative authority)
- Root SDR user (specific secure domain router administrative authority)
- Secure domain router user (specific secure domain router 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](#)” 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 Users](#)” 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 Users](#)” 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](#)” section.)

Secure Domain Router Users

A secure domain router user has restricted access to an SDR as determined by the root-system user or root SDR user. The secure domain router 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 secure domain router user belongs. (See the “[User Groups](#)” section.)

User Groups

Cisco IOS XR software allows the system administrator to configure groups of users and the job characteristics that are common in groups of users. Groups must be explicitly assigned to users. Users are not assigned to groups by default. A user can be assigned to more than one group.

A *user group* defines a collection of users that share a set of attributes, such as access privileges. Each user may be associated with one or more user groups. User groups have the following attributes:

- 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](#)” 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](#)” 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, and change in group B affects group A, even if the group is not re-inherited explicitly.

Task Groups

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

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

Predefined Task Groups

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

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

User-Defined Task Groups

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

Group Inheritance

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

Cisco IOS XR Software Administrative Model

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

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

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

Administrative Access

Administrative access to the system can be lost if the following operations are not well understood and carefully planned. A lockout of all root-system users is a serious issue that requires a system reload to recover the password.

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



Note

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

- Removing the flash card from disk0:, or a disk corruption, may deny auxiliary port authentication, which can affect certain system debugging abilities. However, if the console is available, the system is still accessible.
- Configuring command authorization or EXEC authorization on the console should be done with extreme care, because TACACS+ servers may not be available or may deny every command, which locks the user out. This lockout can occur particularly if the authentication was done with a user not known to the TACACS+ server, or if the TACACS+ user has most or all the commands denied for one reason or another.

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

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

AAA Database

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

Local Database

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



Note

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

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



Note

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

Remote Database

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

Remote AAA Configuration

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

Client Configuration

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

User Groups

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

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

Task Groups

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

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

AAA Configuration

This section provides information about AAA configuration.

Method Lists

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

Rollover Mechanism

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

The following methods are available:

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

Server Grouping

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

Authentication

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

Authentication of Root System User

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

Authentication of Non-Owner Secure Domain Router User

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

Authentication of Owner Secure Domain Router User

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

Authentication of Secure Domain Router User

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

Authentication Flow of Control

AAA performs authentication according to the following process:

1. A user requests authentication by providing a username and password (or secret).
2. AAA verifies the user's password and rejects the user if the password does not match what is in the database.
3. AAA determines the role of the user (root system user, root SDR user, or SDR user).
 - If the user has been configured as a member of a root-system user group, then AAA authenticates the user as a root-system user.
 - If the user has been configured as a member of an owner secure domain router user group, then AAA authenticates the user as an owner secure domain router user.
 - If the user has not been configured as a member of a root-system user group or an owner secure domain router user group, AAA authenticates the user as a secure domain router user.

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

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

Korn Shell Authentication

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

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

- The `ksh` run from the console (using the `run` command) 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:/ ...
```

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 (plus alias also dont require any Task ID permissions). You cannot perform a config replace unless root-lr permissions is 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 [Table 2](#).

Table 2 Task ID Classes

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

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

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

Multiple task ID operations separated by a comma (for example, read, read/write) 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 none specified, the command is used without previous user association to a task ID and operation. In addition, users do not need 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 submenu. For example, to execute the **show redundancy** command, a user needs to be associated to the system (read) task ID and operations as shown in the following example:

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

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/RP0/CPU0:router# admin
RP/0/RP0/CPU0:router(admin)# show redundancy
```

Task IDs for TACACS+ and RADIUS Authenticated Users

Cisco IOS XR 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](#)” section for more details.
- Specify the privilege level in the configuration file of the external TACACS+ and RADIUS servers.
See the “[Privilege Level Mapping](#)” 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 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 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 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:

-
- 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:


```
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:

```
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:

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

Task:      basic-services  :READ   WRITE   EXECUTEDEBUG
Task:      bgp             :READ   WRITE   EXECUTE
Task:      cdp             :READ
Task:      diag            :READ
```

```
Task:          ext-access  :READ          EXECUTE
Task:          logging    :READ
```

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

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

Privilege Level Mapping

For compatibility with TACACS+ daemons that do not support the concept of task IDs, AAA supports a mapping between privilege levels defined for the user in the external TACACS+ server configuration file and local user groups. Following TACACS+ authentication, the task map of the user group that has been mapped from the privilege level returned from the external TACACS+ server is assigned to the user. For example, if a privilege level of 5 is returned from the external TACACS server, AAA attempts to get the task map of the local user group priv5. This mapping process is similar for other privilege levels from 1 to 13. For privilege level 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 want to support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions and to efficiently manage the use of shared resources to offer differing service-level agreements.

Network Security Situations in Which RADIUS is Unsuitable

RADIUS is not suitable in the following network security situations:

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

RADIUS Operation

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

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

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

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

How to Configure AAA Services

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

- [Configuring Task Groups, page SC-19](#) (required)
- [Configuring User Groups, page SC-21](#) (required)
- [Configuring Users, page SC-23](#) (required)
- [Configuring Router to RADIUS Server Communication, page SC-25](#) (optional)
- [Configuring RADIUS Dead-Server Detection, page SC-29](#) (optional)

- [Configuring Per VRF AAA, page SC-31](#) (optional)
- [Configuring a TACACS+ Server, page SC-33](#) (optional)
- [Configuring RADIUS Server Groups, page SC-36](#) (optional)
- [Configuring TACACS+ Server Groups, page SC-38](#) (optional)
- [Configuring AAA Method Lists, page SC-39](#) (required)
- [Applying Method Lists for Applications, page SC-51](#) (required)
- [Configuring Login Parameters, page SC-55](#) (required)

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 selected from the Cisco CRS-1 set of available task IDs. The first configuration task in setting up an the Cisco CRS-1 authorization scheme is 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.

The **inherit taskgroup** command may be used to derive permissions from another group. Circular references are detected and rejected. It is not possible to inherit from the root-system and owner-sdr predefined groups.

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.

Prerequisites

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.

Restrictions

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. **inherit taskgroup** *taskgroup-name*
5. **task** { **read** | **write** | **execute** | **debug** } *taskid-name*

6. Repeat Step 5 for each task ID to be associated with the task group named in Step 2.
7. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	taskgroup <i>taskgroup-name</i> Example: RP/0/RP0/CPU0:router(config)# taskgroup beta	Creates a name for a particular task group and enters task group configuration submenu. <ul style="list-style-type: none"> Specific task groups can be removed from the system by specifying the no form of the taskgroup command.
Step 3	description <i>string</i> Example: RP/0/RP0/CPU0:router(config-tg)# description this is a sample task group description	(Optional) Creates a description of the task group named in Step 2 .
Step 4	inherit taskgroup <i>taskgroup-name</i> Example: RP/0/RP0/CPU0:router(config-tg)# inherit taskgroup sysadmin	(Optional) Derives permissions from another task group and assigns them to the task group named in Step 2 . <ul style="list-style-type: none"> Circular references are detected and rejected. To explicitly define permissions for the task group named in Step 2, omit Step 4 and go to Step 5.
Step 5	task { read write execute debug } <i>taskid-name</i> Example: RP/0/RP0/CPU0:router(config-tg)# task read bgp	Specifies a task ID to be associated with the task group named in Step 2 . <ul style="list-style-type: none"> Assigns read permission for any CLI or API invocations associated with that task ID and performed by a member of the task group. Specific task IDs can be removed from a task group by specifying the no prefix for the task command.

	Command or Action	Purpose
Step 6	Repeat Step 5 for each task ID to be associated with the task group named in Step 2 .	—
Step 7	<pre>end or commit</pre> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-tg)# end or RP/0/RP0/CPU0:router(config-tg)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

What to Do Next

After completing configuration of a full set of task groups, configure a full set of user groups as described in the [“Configuring User Groups”](#) section.

Configuring User Groups

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

Use the **inherit usergroup** command to inherit permissions from other user groups. The user group is inherited by the parent group and forms a union of all task IDs specified in those groups. Circular inclusions are detected and rejected.

Restrictions

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

- configure**
- usergroup** *usergroup-name*

3. **description** *string*
4. **inherit usergroup** *usergroup-name*
5. **taskgroup** *taskgroup-name*
6. Repeat Step 5 for each task group to be associated with the user group named in Step 2.
7. **end**
or
commit

DETAILED STEPS

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

	Command or Action	Purpose
Step 6	Repeat Step 5 for each task group to be associated with the user group named in Step 2.	—
Step 7	<pre>end or commit</pre> <p>Example: RP/0/RP0/CPU0:router(config-ug)# end OR RP/0/RP0/CPU0:router(config-ug)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

What to Do Next

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

Configuring Users

Perform this task to configure a user.

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

SUMMARY STEPS

- configure**
- username** *user-name*
- password** {0 | 7} *password*
or
secret {0 | 5} *secret*
- group** *group-name*
- Repeat Step 4. for each user group to be associated with the user specified in Step 2.

6. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	username <i>user-name</i> Example: RP/0/RP0/CPU0:router(config)# username user1	Creates a name for a new user (or identifies a current user) and enters username configuration submode. <ul style="list-style-type: none"> The <i>user-name</i> argument can be only one word. Spaces and quotation marks are not allowed.
Step 3	password {0 7} <i>password</i> or secret {0 5} <i>secret</i> Example: RP/0/RP0/CPU0:router(config-un)# password 0 pwd1 or RP/0/RP0/CPU0:router(config-un)# secret 0 sec1	Specifies a password for the user named in Step 2. <ul style="list-style-type: none"> Use the secret command to create a secure login password for the user names specified in Step 2. 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. Entering 0 following the secret command specifies that a secure unencrypted (clear-text) password follows. Entering 5 following the secret command specifies that a secure encrypted password follows. Type 0 is the default for the password and secret commands.
Step 4	group <i>group-name</i> Example: RP/0/RP0/CPU0:router(config-un)# group sysadmin	Assigns the user named in Step 2 to a user group that has already been defined through the usergroup command. <ul style="list-style-type: none"> The user takes on all attributes of the user group, as defined by that user group's association to various task groups. Each user must be assigned to at least one user group. A user may belong to multiple user groups.

	Command or Action	Purpose
Step 5	Repeat Step 4 for each user group to be associated with the user specified in Step 2 .	—
Step 6	<pre>end or commit</pre> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-un)# end or RP/0/RP0/CPU0:router(config-un)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

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”](#) or [“Configuring a TACACS+ Server”](#) 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. through Step 6. for each external server to be configured.
8. **end**
or
commit
9. **show radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RP0/CPU0:router# configure</p>	Enters global configuration mode.
Step 2	<p>radius-server host {hostname ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]</p> <p>Example: RP/0/RP0/CPU0:router(config)# radius-server host host1</p>	<p>Specifies the hostname or IP address of the remote RADIUS server host.</p> <ul style="list-style-type: none"> 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. <p>Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius-server host command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.</p>
Step 3	<p>radius-server retransmit retries</p> <p>Example: RP/0/RP0/CPU0:router(config)# radius-server retransmit 5</p>	<p>Specifies the number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up.</p> <ul style="list-style-type: none"> In the example, the number of retransmission attempts is set to 5.
Step 4	<p>radius-server timeout seconds</p> <p>Example: RP/0/RP0/CPU0:router(config)# radius-server timeout 10</p>	<p>Sets the number of seconds a router waits for a server host to reply before timing out.</p> <ul style="list-style-type: none"> In the example, the interval timer is set to 10 seconds.

	Command or Action	Purpose
Step 5	<p>radius-server key {0 <i>clear-text-key</i> 7 <i>encrypted-key</i> <i>clear-text-key</i>}</p> <p>Example: RP/0/RP0/CPU0:router(config)# radius-server key 0 samplekey</p>	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 6	<p>radius source-interface <i>type instance</i> [vrf <i>vrf-id</i>]</p> <p>Example: RP/0/RP0/CPU0:router(config)# radius source-interface POS 0/3/0/1</p>	<p>(Optional) Forces RADIUS to use the IP address of a specified interface or subinterface for all outgoing RADIUS packets.</p> <ul style="list-style-type: none"> The specified interface or subinterface must have an IP address associated with it. If the specified interface or subinterface does not have an IP address or is in the down state, then RADIUS reverts to the default. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state. <p>The vrf keyword enables the specification on a per-VRF basis.</p>
Step 7	Repeat Step 2 through Step 6 for each external server to be configured.	—
Step 8	<p>end or commit</p> <p>Example: RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	<p>show radius</p> <p>Example: RP/0/RP0/CPU0:router# show radius</p>	(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](#)” 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 downtime 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. **end**
or
commit
6. **show radius dead-criteria host** *ip-addr* [**auth-port** *auth-port*] [**acct-port** *acct-port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	radius-server deadtime <i>minutes</i> Example: RP/0/RP0/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 <i>seconds</i> Example: RP/0/RP0/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 <i>tries</i> Example: RP/0/RP0/CPU0:router(config)# radius-server dead-criteria tries 4	Establishes the number of tries for the dead-criteria conditions for a RADIUS server to be marked as dead.
Step 5	end or commit Example: RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit	Saves configuration changes. <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	show radius dead-criteria host <i>ip-addr</i> [auth-port <i>auth-port</i>] [acct-port <i>acct-port</i>] Example: RP/0/RP0/CPU0:router# show radius dead-criteria host 172.19.192.80	(Optional) Displays dead-server-detection information that has been requested for a RADIUS server at the specified IP address.

Configuring Per VRF AAA

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

New Vendor-Specific Attributes (VSAs)

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

The Cisco IOS XR 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 TACACS+ specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes. This definition allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

Table 3 describes the VSAs that are now supported for Per VRF AAA.

Table 3 Supported VSAs¹ for Per VRF AAA

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

1. The RADIUS VSAs—rad-serv, rad-serv-source-if, and rad-serv-vrf—must have the prefix “aaa:” before the VSA name.

SUMMARY STEPS

1. **configure**
2. **aaa group server radius group-name**

3. **server-private** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
4. **vrf** *vrf-name*
5. **end**
or
commit

DETAILED STEPS

Command or Action	Purpose
Step 1 configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2 aaa group server radius <i>group-name</i> Example: RP/0/RP0/CPU0:router(config)# aaa group server radius radgroup1 RP/0/RP0/CPU0:router(config-sg-radius)#	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3 server-private { <i>hostname</i> <i>ip-address</i> } [auth-port <i>port-number</i>] [acct-port <i>port-number</i>] [timeout <i>seconds</i>] [retransmit <i>retries</i>] [key <i>string</i>] Example: RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 timeout 5 RP/0/RP0/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.

Command or Action	Purpose
<p>Step 4 <code>vrf vrf-name</code></p> <p>Example: RP/0/RP0/CPU0:router(config-sg-radius)# vrf v2.44.com</p>	<p>Configures the VRF reference of an AAA RADIUS server group.</p> <p>Note Private server IP addresses can overlap with those configured globally and the VRF definitions can help to distinguish them.</p>
<p>Step 5 <code>end</code> OR <code>commit</code></p> <p>Example: RP/0/RP0/CPU0:router(config-sg-radius)# end OR RP/0/RP0/CPU0:router(config-sg-radius)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

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 the TACACS+ servers. The timeout specifies how long the AAA server waits to receive a response from the TACACS+ server. The key specifies an authentication and encryption key shared between the AAA server and the TACACS+ server.

SUMMARY STEPS

- configure**
- tacacs-server host** *host-name* **port** *port-number*
- tacacs-server host** *host-name* **timeout** *seconds*
- tacacs-server host** *host-name* **key** [**0** | **7**] *auth-key*
- tacacs-server host** *host-name* **single-connection**
- tacacs source-interface** *type instance*

7. Repeat Step 2. through Step 5. for each external server to be configured.
8. **end**
or
commit
9. **show tacacs**

DETAILED STEPS

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

	Command or Action	Purpose
Step 5	<p>tacacs-server host <i>host-name</i> single-connection</p> <p>Example: RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 single-connection</p>	Prompts the router to multiplex all TACACS+ requests to this server over a single TCP connection. By default, a separate connection is used for each session.
Step 6	<p>tacacs source-interface <i>type instance</i></p> <p>Example: RP/0/RP0/CPU0:router(config)# tacacs source-interface POS 0/4/0/0</p>	<p>(Optional) Specifies the source IP address of a selected interface for all outgoing TACACS+ packets.</p> <ul style="list-style-type: none"> The specified interface or subinterface must have an IP address associated with it. If the specified interface or subinterface does not have an IP address or is in the down state, then TACACS+ reverts to the default. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state.
Step 7	Repeat Step 2. through Step 5. for each external server to be configured.	—
Step 8	<p>end OR commit</p> <p>Example: RP/0/RP0/CPU0:router(config)# end OR RP/0/RP0/CPU0:router(config)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	<p>show tacacs</p> <p>Example: RP/0/RP0/CPU0:router# show tacacs</p>	(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](#)” 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](#)” section.)

Prerequisites

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** {*host-name* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*]
4. Repeat Step 3. for every external server to be added to the server group named in Step 2.
5. **server-private** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
6. **deadtime** *minutes*
7. **end**
or
commit
8. **show radius server-groups** [*group-name* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	aaa group server radius <i>group-name</i> Example: RP/0/RP0/CPU0:router(config)# aaa group server radius radgroup1	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	server { <i>hostname</i> <i>ip-address</i> } [auth-port <i>port-number</i>] [acct-port <i>port-number</i>] Example: RP/0/RP0/CPU0:router(config-sg-radius)# server 192.168.20.0	Specifies the hostname or IP address of an external RADIUS server. <ul style="list-style-type: none"> • After the server group is configured, it can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).

Command or Action	Purpose
<p>Step 4 Repeat Step 3 for every external server to be added to the server group named in Step 2.</p>	<p>—</p>
<p>Step 5</p> <pre>server-private {hostname ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]</pre> <p>Example: RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.10.130.2 auth-port 1600 acct-port 1666 key code</p>	<p>Configures the IP address of the private RADIUS server for the group server.</p> <p>Note If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.</p>
<p>Step 6</p> <pre>deadtime minutes</pre> <p>Example: RP/0/RP0/CPU0:router(config-sg-radius)# deadtime 1</p>	<p>Configures the deadtime value at the RADIUS server group level.</p> <ul style="list-style-type: none"> The <i>minutes</i> argument specifies the length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 (24 hours). The range is from 1 to 1440. <p>The example specifies a one-minute deadtime for RADIUS server group radgroup1 when it has failed to respond to authentication requests for the deadtime command</p> <p>Note You can configure the group-level deadtime after the group is created.</p>
<p>Step 7</p> <pre>end or commit</pre> <p>Example: RP/0/RP0/CPU0:router(config-sg-radius)# end or RP/0/RP0/CPU0:router(config-sg-radius)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
<p>Step 8</p> <pre>show radius server-groups [group-name [detail]]</pre> <p>Example: RP/0/RP0/CPU0:router# show radius server-groups</p>	<p>(Optional) Displays information about each RADIUS server group that is configured in the system.</p>

What to Do Next

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

Configuring TACACS+ Server Groups

This task configures TACACS+ server groups.

The user 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](#)” section.)

Prerequisites

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

SUMMARY STEPS

1. **configure**
2. **aaa group server tacacs+ group-name**
3. **server {host-name | ip-address}**
4. Repeat Step 3. for every external server to be added to the server group named in Step 2.
5. **end**
or
commit
6. **show tacacs server-groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	aaa group server tacacs+ group-name Example: RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1	Groups different server hosts into distinct lists and enters the server group configuration mode.
Step 3	server {hostname ip-address} Example: RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 192.168.100.0	Specifies the hostname or IP address of an external TACACS+ server. <ul style="list-style-type: none"> • When configured, this group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).

	Command or Action	Purpose
Step 4	Repeat Step 3 for every external server to be added to the server group named in Step 2 .	—
Step 5	<pre>end or commit</pre> <p>Example: RP/0/RP0/CPU0:router(config-sg-tacacs+)# end OR RP/0/RP0/CPU0:router(config-sg-tacacs+)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	<pre>show tacacs server-groups</pre> <p>Example: RP/0/RP0/CPU0:router# show tacacs server-groups</p>	(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 by configuring authentication, authorization, and accounting. (See the [“Configuring AAA Method Lists”](#) section.)

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, page SC-40](#) (required)
- [Configuring Authorization Method Lists, page SC-42](#) (required)
- [Configuring Accounting Method Lists, page SC-46](#) (required)

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.

Restrictions

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.



Note

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. **end**
or
commit
4. Repeat Step 1 through Step 3 for every authentication method list to be configured.

DETAILED STEPS

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

	Command or Action (continued)	Purpose (continued)
Step 3	<pre>end or commit</pre> <p>Example: RP/0/RP0/CPU0:router(config)# end OR RP/0/RP0/CPU0:router(config)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
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”](#) 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 Cisco IOS XR software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS XR software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or until all methods defined have been exhausted.

**Note**

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

Method lists are specific to the type of authorization being requested. Cisco IOS XR software supports four types of AAA authorization:

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

**Note**

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

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

**Note**

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

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

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

“Command” authorization, as a result of adding a command authorization method list to a line template, is separate from, and 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+ or RADIUS 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. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	aaa authorization {commands eventmanager exec network} {default list-name} {none local group {tacacs+ radius group-name}} Example: RP/0/RP0/CPU0:router(config)# aaa authorization commands listname1 group tacacs+	Creates a series of authorization methods, or a method list. <ul style="list-style-type: none"> • The commands keyword configures authorization for all EXEC shell commands. Command authorization applies to the EXEC mode commands issued by a user. Command authorization attempts authorization for all EXEC mode commands. • The eventmanager keyword applies an authorization method for authorizing an event manager (fault manager). • The exec keyword configures authorization for an interactive (EXEC) 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.

Command or Action (continued)	Purpose (continued)
	<ul style="list-style-type: none"> • A <i>list-name</i> character string identifies the authorization method list. The method list itself follows the method list name. Method list types are entered in the preferred sequence. The listed method list types can be any one of the following: <ul style="list-style-type: none"> – none—The network access server (NAS) does not request authorization information. Authorization always succeeds. No subsequent authorization methods will be attempted. However, the task ID authorization is always required and cannot be disabled. – local—Uses local database for authorization.
	<ul style="list-style-type: none"> – group tacacs+—Uses the list of all configured TACACS+ servers for authorization. The NAS exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating AV pairs, which are stored in a database on the TACACS+ security server, with the appropriate user. – group radius—Uses the list of all configured RADIUS servers for authorization. – group 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.
<p>Step 3</p> <p>end or commit</p> <p>Example: RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> – Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

What to Do Next

After configuring authorization method lists, configure accounting method lists. (See the “[Configuring Accounting Method Lists](#)” 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. **aaa accounting** { **commands** | **exec** | **network** } { **default** | *list-name* } { **start-stop** | **stop-only** } { **none** | *method* }
3. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RP0/CPU0:router# configure</p>	Enters global configuration mode.
Step 2	<p>aaa accounting {commands exec network} {default <i>list-name</i>} {start-stop stop-only} {none <i>method</i>}</p> <p>Example: RP/0/RP0/CPU0:router(config)# aaa accounting commands default stop-only group tacacs+</p>	<p>Creates a series of accounting methods, or a method list.</p> <ul style="list-style-type: none"> • The commands keyword enables accounting for EXEC shell commands. • The exec keyword enables accounting for an interactive (EXEC) session. • The network keyword enables accounting for all network-related service requests, such as Point-to-Point Protocol (PPP). • The default keyword causes the listed accounting methods that follow this keyword to be the default list of methods for accounting. • A <i>list-name</i> character string identifies the accounting method list. • The start-stop keyword sends a “start accounting” notice at the beginning of a process and a “stop accounting” notice at the end of a process. The requested user process begins regardless of whether the “start accounting” notice was received by the accounting server. • The stop-only keyword sends a “stop accounting” notice at the end of the requested user process. • The none keyword states that no accounting is performed. • The method list itself follows the start-stop keyword. Method list types are entered in the preferred sequence. The method argument lists the following types: <ul style="list-style-type: none"> – group tacacs+—Use the list of all configured TACACS+ servers for accounting. – group radius—Use the list of all configured RADIUS servers for accounting. – group group-name—Use a named server group, a subset of TACACS+ or RADIUS servers for accounting as defined by the aaa group server tacacs+ or aaa group server radius command. • The example defines a default command accounting method list, in which accounting services are provided by a TACACS+ security server, with a stop-only restriction.

Command or Action (continued)	Purpose (continued)
<p>Step 3</p> <pre>end or commit</pre> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

What to Do Next

After configuring method lists, apply those method lists. (See the [“Applying Method Lists for Applications”](#) 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

- configure**
- aaa accounting update {newinfo | periodic *minutes*}**
- end**
or
commit

DETAILED STEPS

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

Command or Action (continued)	Purpose (continued)
<p>Step 2 <code>aaa accounting update {newinfo periodic minutes}</code></p> <p>Example: RP/0/RP0/CPU0:router(config)# <code>aaa accounting update periodic 30</code></p>	<p>Enables periodic interim accounting records to be sent to the accounting server.</p> <ul style="list-style-type: none"> • If the newinfo keyword is used, interim accounting records are sent to the accounting server every time 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. <p> Caution The periodic keyword causes heavy congestion when many users are logged in to the network.</p>
<p>Step 3 <code>end</code> or <code>commit</code></p> <p>Example: RP/0/RP0/CPU0:router(config)# <code>end</code> or RP/0/RP0/CPU0:router(config)# <code>commit</code></p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> – Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

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, page SC-51](#) (required)
- [Enabling Accounting Services, page SC-53](#) (required)

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. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	line {aux console default template <i>template-name</i> }	Enters line template configuration mode.
	Example: RP/0/RP0/CPU0:router(config)# line console	

Command or Action (continued)	Purpose (continued)
<p>Step 3</p> <pre>authorization {commands exec} {default list-name}</pre> <p>Example: RP/0/RP0/CPU0:router(config-line)# authorization commands listname5</p>	<p>Enables AAA authorization for a specific line or group of lines.</p> <ul style="list-style-type: none"> The commands keyword enables authorization on the selected lines for all commands. The exec keyword enables authorization for an interactive (EXEC) 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.
<p>Step 4</p> <pre>end or commit</pre> <p>Example: RP/0/RP0/CPU0:router(config-line)# end or RP/0/RP0/CPU0:router(config-line)# commit </p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

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”](#) section.)

Enabling Accounting Services

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RP0/CPU0:router# configure</p>	Enters global configuration mode.
Step 2	<p>line {aux console default template template-name}</p> <p>Example: RP/0/RP0/CPU0:router(config)# line console</p>	Enters line template configuration mode.
Step 3	<p>accounting {commands exec} {default list-name}</p> <p>Example: RP/0/RP0/CPU0:router(config-line)# accounting commands listname7</p>	<p>Enables AAA accounting for a specific line or group of lines.</p> <ul style="list-style-type: none"> • The commands keyword enables accounting on the selected lines for all EXEC shell commands. • The exec keyword enables accounting for an interactive (EXEC) 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	<p>end or commit</p> <p>Example: RP/0/RP0/CPU0:router(config-line)# end or RP/0/RP0/CPU0:router(config-line)# commit</p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> – Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

What to Do Next

After applying accounting method lists by enabling AAA accounting services, configure login parameters. (See the “[Configuring Login Parameters](#)” 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. **end**
or
commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	line template <i>template-name</i> Example: RP/0/RP0/CPU0:router(config)# line template alpha	Specifies a line to configure and enters line template configuration mode.

	Command or Action (continued)	Purpose (continued)
Step 3	<pre>timeout login response seconds</pre> <p>Example: RP/0/RP0/CPU0:router(config-line)# timeout login response 20 </p>	<p>Sets the interval that the server waits for reply to a login.</p> <ul style="list-style-type: none"> The <i>seconds</i> argument specifies the timeout interval (in seconds) from 0 to 300. The default is 30 seconds. The example shows how to change the interval timer to 20 seconds.
Step 4	<pre>end</pre> <p>OR</p> <pre>commit</pre> <p>Example: RP/0/RP0/CPU0:router(config-line)# end</p> <p>OR</p> <pre>RP/0/RP0/CPU0:router(config-line)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Configuring AAA Services

This section provides the following configuration example:

- [Configuring AAA Services: Example, page SC-56](#)

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-system user. Configure similar settings for username user2.

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

username user2
secret lab
exit
```

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

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

usergroup uga
taskgroup tga
description usergroup uga
exit
```

Username user2 is configured to inherit from user group uga.

```
username user2
group uga
exit
```

Three TACACS servers are configured.

```
tacacs-server host 1.1.1.1 port 1 key abc
tacacs-server host 2.2.2.2 port 2 key def
tacacs-server host 3.3.3.3 port 3 key ghi
```

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

```
usergroup priv5
taskgroup operator
exit
```

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

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

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

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

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

```
aaa default-taskgroup tga
```

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

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

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

```
aaa group server tacacs+ abc
server 3.3.3.3
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	<i>Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software</i> module in <i>Cisco IOS XR System Security Command Reference</i>

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

