



CHAPTER 3

Configuring the Guard

This chapter describes how to configure the Cisco Guard (Guard) services.

This chapter refers to the Cisco Detector (Detector), the companion product of the Guard. The Detector is a Distributed Denial of Service (DDoS) attack detection device that analyzes a copy of the zone traffic. The Detector can activate the Guard attack mitigation services when the Detector determines that the zone is under attack. The Detector can also synchronize zone configurations with the Guard. For more information about the Detector, see the *Cisco Traffic Anomaly Detector Module Configuration Guide* and *Cisco Traffic Anomaly Detector Configuration Guide*.

This chapter contains the following sections:

- [Activating Guard Services](#)
- [Configuring Access Control Using AAA](#)
- [Establishing Communication with the Detector](#)
- [Configuring a Date and a Time](#)
- [Synchronizing the Guard Clock with an NTP Server](#)
- [Managing SSH Keys](#)
- [Configuring the Keys for SFTP and SCP Connections](#)
- [Changing the Hostname](#)
- [Enabling SNMP Traps](#)
- [Configuring SNMP Community Strings](#)
- [Configuring the Login Banner](#)
- [Configuring the WBM Logo](#)
- [Configuring the Session Timeout](#)

Activating Guard Services

The Guard has several service options which you can activate by enabling the service and then defining the IP address that is permitted access to the service. With the exception of the Secure Shell service which is always active, this section describes how to activate the services.

The Guard services are as follows:

- Internode communication service—The Guard uses this service when establishing a communication channel with the Cisco Traffic Anomaly Detector. See the [“Establishing Communication with the Detector” section on page 3-17](#) for more information.
- Network Time Protocol (NTP) service—The Guard provides a time synchronization service that allows you to synchronize the Guard with a time synchronization server. To enable time synchronization, you must configure an NTP server. See the [“Synchronizing the Guard Clock with an NTP Server” section on page 3-22](#) for more information.
- Simple Network Management Protocol (SNMP) server service—You can access the Guard using SNMP to retrieve information as defined by the following MIBs:
 - Riverhead private MIB
 - MIB2 (RFC1213-MIB)—All of the MIB groups with the exceptions of the EGP and transmission MIB groups.
 - UCDAVIS (UCD-SNMP-MIB)—Only the following MIB groups: memory, latable, systemStats, version, and snmperrs.

See the MIB file that is released with the software version for information about the MIB definitions.



Note The Riverhead MIB contains 64-bit counters. To read the MIB, you must use a browser that supports SNMP version 2.

- SNMP trap service—When you activate the snmp-trap service, the Guard generates SNMP traps. See the [“Enabling SNMP Traps” section on page 3-26](#) for more information.
- Secure Shell service—The SSH service is always active. See the [“Accessing the Guard with SSH” section on page 2-17](#) and the [“Managing SSH Keys” section on page 3-23](#) for more information.
- Web-Based Manager (WBM) service—You can monitor and control the Guard from the web using a web browser. See the [“Managing the Guard with a Web-Based Manager” section on page 2-15](#) for more information.
- MultiDevice Manager (MDM) service—Using a web browser, you can monitor and control the Guard and other Guard and Detector devices from the MDM server. See the [“Managing the Guard with the Cisco DDoS MultiDevice Manager” section on page 2-16](#) for more information.

By default, all Guard services, except SSH, are disabled.

To activate a Guard service, perform the following steps:

Step 1 Enable the Guard service by entering the following command in configuration mode:

```
service {internode-comm | mdm | ntp | snmp-server | snmp-trap | wbm}
```

Table 3-1 provides the keywords for the **service** command.

Table 3-1 Keywords for the **service** Command

| Service | Description |
|-----------------------|--|
| internode-comm | Specifies the internode communication service. |
| mdm | Specifies the MDM service. |
| ntp | Specifies the NTP service. |
| snmp-server | Specifies the SNMP server service. |
| snmp-trap | Specifies the SNMP trap service. |
| wbm | Specifies the WBM service. |

Step 2 Permit access to the Guard service by entering one of the following commands:

- For the MDM service, permit access to the Guard service from the MDM by entering the following command in configuration mode:

```
mdm server ip-addr
```


The *ip-addr* argument defines the IP address of your MDM server. Enter the IP address in dotted-decimal notation.

- For all other services, permit access to the Guard service and enable connectivity by entering the following command in configuration mode:

```
permit {internode-comm | ntp | snmp-server | ssh | wbm}
{* | ip-address-general [ip-mask]}
```

Table 3-2 provides the arguments and keywords for the **permit** command.

Table 3-2 Arguments and Keywords for the **permit** Command

| Parameter | Description |
|---------------------------|--|
| internode-comm | Specifies the internode communication service. |
| ntp | Specifies the NTP service. |
| snmp-server | Specifies the SNMP server service. |
| ssh | Specifies the SSH service. |
| wbm | Specifies the WBM service. |
| * | Wildcard character that permits access to a service by all device IP addresses. |
| |  <p>Caution For security reasons, we do not recommend that you permit access to a service from all IP addresses.</p> |
| <i>ip-address-general</i> | IP address from which to permit access. Enter the IP address in dotted-decimal notation (for example, enter 192.168.10.2). |
| <i>ip-mask</i> | (Optional) IP subnet mask. Enter the subnet mask in dotted-decimal notation (for example, enter 255.255.255.0). The default subnet mask is 255.255.255.255. |

The following example shows how to activate the WBM service:

```
user@GUARD-conf# service wbm
user@GUARD-conf# permit wbm 192.168.10.35
```

Configuring Access Control Using AAA

Authentication, Authorization, and Accounting (AAA) is a method for controlling user access to the Guard and the Guard services. AAA provides the following features:

- **Authentication**—Identifies a user before the user is allowed access to the system and system services.
- **Authorization**—Determines what a user is allowed to perform once access to the system is obtained. This process occurs after the user is authenticated.
- **Accounting**—Records what a user is performing or has performed. Accounting allows you to track the services that users are accessing.

The Guard is preconfigured with the following system user accounts:

- **admin**—The admin user account is configured with the administration access rights, allowing access to the Guard CLI and all its functionality. When connecting to the Guard CLI for the first time, you are required to set a password for this account. Use the admin user account to configure additional user accounts.
- **riverhead**—The riverhead user account is configured with dynamic access rights. The Guard uses this user account to establish the initial communication channel with the Cisco Traffic Anomaly Detector. When you connect to the Guard CLI for the first time, you are required to set a password for this account.

You cannot delete system user accounts.

You can divide the Guard user community into domains and assign passwords for secure management access. We recommend that you create new user accounts and avoid using the system user accounts after the initial configuration so that you can monitor user actions.

The section contains the following topics:

- [Configuring Authentication](#)
- [Configuring Authorization](#)
- [Configuring Accounting](#)
- [Configuring the TACACS+ Server Attributes](#)

Configuring Authentication

You can configure which authentication method that the Guard uses when a user tries to log into the Guard or requests a higher privilege level (using the **enable** command). The Guard offers the following authentication options:

- **Local authentication**—Uses locally configured login and enable passwords for authentication. This authentication method is the default. See the “[Configuring Local Authentication](#)” section on [page 3-6](#) for more information.
- **Terminal Access Controller Access Control System Plus (TACACS+) authentication**—Remote user authentication using one or more TACACS+ servers.

**Note**

When you define a user to authenticate on the TACACS+ server, you must also define authorization for the user or the user will have access to **show** commands only (see the “[Configuring Authorization](#)” section on page 3-8).

You can configure the Guard to use one or both of the user authentication methods. When using Terminal Access Controller Access-Control System Plus (TACACS+) authentication, you can define multiple TACACS+ servers. Defining more than one authentication method provides a backup in case the initial method fails due to a communication error.

The Guard authenticates a user by using each of the methods that you define and in the order in which you define them on the Guard. To view the list of defined authentication methods, enter the **show running config** command. The Guard attempts to authenticate the user using the first method on the list. If the first authentication method does not respond, the Guard sequentially selects the next authentication method on the list until it finds one that responds.

You can configure the action that the Guard takes when it receives a response from the first TACACS+ server by using the **tacacs-server first-hit** command. If you disable the first-hit option (the default setting) and the first server rejects the authentication, the Guard sequentially scans the other TACACS+ servers to find a server that accepts the authentication. User authentication fails when no defined TACACS+ servers accept the authentication or the Guard cannot communicate with any of the servers. If you enable the first-hit option, the Guard accepts the authentication response (reject or accept) of the first TACACS+ server to respond as the final decision. By default, the first-hit option is disabled. For more information about the **tacacs-server first-hit** command, see the “[Configuring the TACACS+ Search Method](#)” section on page 3-16.

**Note**

You can configure the Guard to use its local database as a fallback for user authentication when the Guard cannot communicate with the TACACS+ servers (see the “[Configuring Authentication Methods](#)” section).

This section contains the following topics:

- [Configuring Authentication Methods](#)
- [Configuring Local Authentication](#)

Configuring Authentication Methods

To configure the authentication method that the Guard uses, perform the following steps:

- Step 1** Configure the TACACS+ server connection if TACACS+ authentication is required. See the “[Configuring the TACACS+ Server Attributes](#)” section on page 3-14 for more information.
- Step 2** Define the authentication method by entering the following command in configuration mode:
- ```
aaa authentication {enable | login} {local | tacacs+} [tacacs+ | local]
```

Table 3-3 provides the keywords for the **aaa authentication** command.

**Table 3-3** Keywords for the *aaa authentication* Command

| Parameter              | Description                                                                               |
|------------------------|-------------------------------------------------------------------------------------------|
| <b>enable</b>          | Allows the Guard to authenticate when a user enters a higher privilege level.             |
| <b>login</b>           | Allows the Guard to authenticate when a user logs in.                                     |
| <b>local</b>           | Specifies the local database that the Guard uses to authenticate a user.                  |
| <b>tacacs+</b>         | Allows a TACACS+ server to authenticate a user.                                           |
| <b>tacacs+   local</b> | (Optional) Specifies an alternative authentication method if the configured method fails. |

If you access the Guard from a console session, it uses the local user database for authentication regardless of the defined authentication method.

To change the authentication method, reenter the command.

The following example shows how to configure authentication on entering a higher privilege level. The primary authentication method is configured to TACACS+, and the secondary authentication method is configured to the local user database.

```
user@GUARD-conf# aaa authentication enable tacacs+ local
```

## Configuring Local Authentication

The Guard initially has a preconfigured username (called a user definition) with administration privileges, which allows you to create new users. A user definition allows you to divide the Guard user community into domains and to assign passwords for secure management access.

To enable authentication of CLI users with a TACACS+ server, see the [“Configuring Authentication” section on page 3-4](#).

This section contains the following topics:

- [Adding a User](#)
- [Changing Your Password](#)
- [Changing the Passwords of Other Users](#)
- [Deleting a User from the Local User Database](#)

### Adding a User

To add a user to the Guard local database, use the following command in configuration mode:

```
username username { admin | config | dynamic | show } [password]
```

Table 3-4 provides the arguments and keywords for the **username** command.

**Table 3-4 Arguments and Keywords for the username Command**

| Parameter       | Description                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>username</i> | Username. A case-sensitive alphanumeric string from 1 to 63 characters that starts with an alphabetic letter. The string cannot contain spaces but can contain underscores.                      |
| <b>admin</b>    | Provides access to all operations.                                                                                                                                                               |
| <b>config</b>   | Provides access to all operations except for operations relating to user definition, deletion, and modification.                                                                                 |
| <b>dynamic</b>  | Provides access to monitoring and diagnostic operations, protection, and learning-related operations. Users with Dynamic privileges can also configure flex-content filters and dynamic filters. |
| <b>show</b>     | Provides access to monitoring and diagnostic operations.                                                                                                                                         |
| <i>password</i> | (Optional) Password associated with the username. Enter an alphanumeric case-sensitive 6- to 24-character string with no spaces. If you do not enter a password, you are prompted for it.        |

The following example shows how to configure a new user and set the password:

```
user@GUARD-conf# username Robbin config 1234
```

Users enter passwords in clear text but the Guard configuration file displays passwords in an encrypted manner. This example displays the Guard configuration file (running-config):

```
username Richard config encrypted 840xdMk3
```

The **encrypted** keyword in the previous example indicates that the password is encrypted.

To display the list of users configured on the Guard, use the **show running-config** or **show guard** commands.

To display a list of the users currently logged into the CLI, use the **show users** command.

## Changing Your Password

You can change your own password. Administrators can change their own password and the passwords of other users (see the [“Changing the Passwords of Other Users”](#) section on page 3-8).

To change your own password, perform the following steps:

- 
- Step 1** Enter the following command in global mode:  
**password**
  - Step 2** Enter your current password. The system prompts you for a new password.
  - Step 3** Enter a new password. The password must be an alphanumeric, 6- to 24-character string with no spaces. The password is case sensitive. The system prompts you to confirm the new password by typing it again.
-

The following example shows how to change your password:

```
user@GUARD# password
Old Password: <old-password>
New Password: <new-password>
Retype New Password: <new-password>
```

## Changing the Passwords of Other Users

You must have administration user privileges to change the password of other users.

To change the password of another user, perform the following steps:

---

**Step 1** Enter the following command in global mode:

```
password username-password
```

The *username-password* argument is the user whose password you are changing.

**Step 2** Enter a new password.

The password must be an alphanumeric, 6- to 24-character string with no spaces. The password is case sensitive. The system prompts you to confirm the new password by typing it again.

---

The following example shows how the administrator changes the password of the user Jose:

```
user@GUARD# password Jose
New Password: <new-password>
Retype New Password: <new-password>
```

## Deleting a User from the Local User Database

When you delete a user from the local user database, the associated user cannot access the Guard if authentication is performed using the local user database only.

To delete a user from the Guard local user database, use the **no username** *username* command.

The following example shows how to delete a user from the local user database:

```
user@GUARD-conf# no username Robbin
```

## Configuring Authorization

You can limit the services available to a user. When you enable authorization, the Guard verifies the user profile, which is located either in the local user database or on a TACACS+ security server. The user is permitted access to the requested service only if the information in the user profile allows it.

You can configure which authorization method that the Guard uses when a user tries to execute a command. The Guard offers the following authorization options:

- TACACS+ authorization—Authorizes users through a TACACS+ server. Access to a subsequent server is initiated, if a server is defined, only if communication to a server fails.

Two types of TACACS+ authorization are supported:

- EXEC authorization—Determines the user privilege level once when the user is authenticated upon logging into the Guard.

- Command authorization—Consults a TACACS+ server to get authorization for each command after the user enters the command.

TACACS+ authorization enables you to specify access rights for each command.

**Caution**

We recommend that you limit authorization to the **copy running-config** command because using the **copy running-config** command allows a user to access all configuration commands, regardless of whether the user is actually authorized every command in the configuration file.

- Local authorization—Uses locally configured user profiles for command group access control. Authorization is defined for all commands at the specified privilege level. This authorization method is the default.

The Guard can use local authorization when communication to the TACACS+ server fails.

You can configure a sequential authorization list that defines the methods for authorizing a user, allows you to designate one or more methods to be used for authorization, and provides a backup if communication to the initial method fails.

The Guard uses the first method that you listed to authorize users; if that method does not respond, the Guard selects the second authorization method. The authorization fails only if both authorization methods do not succeed.

To configure the Guard to consider an authentication rejection as final and stop further searching with other TACACS+ servers or the local user database, you can configure the TACACS+ server attributes. See the “[Configuring the TACACS+ Server Attributes](#)” section on page 3-14 for more information.

This section contains the following topics:

- [Configuring Local Authorization](#)
- [Configuring Authorization Methods](#)
- [Disabling Tab Completion of Zone Names](#)

## Configuring Local Authorization

Access to Guard operations depends on the user privilege level. You can limit the operations available to a user. The Guard checks the user profile to verify the user access rights. Once authorized, the user is granted access to the requested operation only if the information in the user profile allows it. See [Table 2-1](#) for more information about user privilege levels.

This section contains the following topics:

- [Assigning Privilege Levels with Passwords](#)
- [Moving Between User Privilege Levels](#)

### Assigning Privilege Levels with Passwords

You can set passwords that restrict access to user privilege levels. After you specify the privilege level and the password, you can give the password to a user who needs to access this level. Without knowing the privilege level password, the user cannot move to the password-protected level.

To set a local password to control access to a privilege level, use the following command in configuration mode:

```
enable password [level level] [password]
```

Table 3-5 provides the arguments for the **enable password** command.

**Table 3-5 Arguments for the enable password Command**

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>level</b> <i>level</i> | (Optional) Specifies the user privilege level. The level can be one of the following: <ul style="list-style-type: none"> <li>• <b>admin</b>—Provides access to all operations.</li> <li>• <b>config</b>—Provides access to all operations except for operations relating to user definition, deletion, and modification.</li> <li>• <b>dynamic</b>—Provides access to monitoring and diagnostic operations, protection, and learning-related operations. Users with Dynamic privileges can also configure flex-content filters and dynamic filters.</li> <li>• <b>show</b>—Provides access to monitoring and diagnostic operations.</li> </ul> The default level is admin. |
| <b>password</b>           | (Optional) Password for the privilege level. The password must be an alphanumeric, 6- to 24-character string with no spaces. The password is case sensitive. If you do not enter a password, you are prompted for it.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

The following example shows how to assign a password to the user privilege level admin:

```
user@GUARD-conf# enable password level admin <password>
```

## Moving Between User Privilege Levels

Authorized users can move between user privilege levels.

To move between user privilege levels, perform the following steps:

**Step 1** Enter the following command in global mode:

```
enable [level]
```

Table 3-6 provides the values for the optional *level* argument which specifies the user privilege level.

**Table 3-6 Keywords for the enable Command**

| Parameter      | Description                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>admin</b>   | Provides access to all operations. This is the default.                                                                                                                                          |
| <b>config</b>  | Provides access to all operations except for operations relating to user definition, deletion, and modification                                                                                  |
| <b>dynamic</b> | Provides access to monitoring and diagnostic operations, protection, and learning-related operations. Users with Dynamic privileges can also configure flex-content filters and dynamic filters. |
| <b>show</b>    | Provides access to monitoring and diagnostic operations.                                                                                                                                         |

**Step 2** Enter the privilege level password.

The following example shows how to switch to the admin privilege level:

```
user@GUARD> enable admin
Enter enable admin Password: <password>
```

To return to the show privilege level (as described in [Table 3-5](#)), use the **disable** command.

## Configuring Authorization Methods


To configure the authorization method, perform the following steps:

- Step 1** Configure the TACACS+ server connection if TACACS+ authorization is required. See the “[Configuring the TACACS+ Server Attributes](#)” section on [page 3-14](#) for more information.
- Step 2** Define the authorization method by entering one of the following commands in configuration mode:
- **aaa authorization exec tacacs+**
  - **aaa authorization commands level tacacs+**

To remove an authorization method, use the **no** form of the command.

[Table 3-7](#) provides the arguments and keywords for the **aaa authorization** command.

**Table 3-7 Arguments and Keywords for the aaa authorization Command**

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>exec</b>     | Runs authorization to determine if the user is allowed to run an EXEC shell. The Guard consults the TACACS+ server to determine the privilege level for an authenticated user.<br><br><div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p><b>Caution</b> You must configure authentication for the user on a TACACS+ server before configuring user authorization or the user may not be able to access the Guard module (see the “<a href="#">Configuring Authorization</a>” section on <a href="#">page 3-8</a>).</p> </div> |
| <b>commands</b> | Runs authorization for all commands at the specified privilege level. To configure authorization for more than one privilege level, use the command for each privilege level that requires authorization.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>level</i>    | Authorization for the specified privilege level. The level can be one of the following: <ul style="list-style-type: none"> <li>• <b>admin</b>—Provides access to all operations.</li> <li>• <b>config</b>—Provides access to all operations except for operations relating to user definition, deletion, and modification.</li> <li>• <b>dynamic</b>—Provides access to monitoring and diagnostic operations, protection, and learning-related operations. Users with Dynamic privileges can also configure flex-content filters and dynamic filters.</li> </ul>                                                                         |
| <b>tacacs+</b>  | Verifies the user access rights with a TACACS+ server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

We recommend that you do not configure authorization for **show** privilege level commands because it may affect Guard performance.

**Note**


---

No TACACS+ authorization is performed for commands that you enter from console sessions.

---

The following example shows how to configure authorization for commands that require the config privilege level:

```
user@GUARD-conf# aaa authorization commands config tacacs+
```

**Caution**


---

You must grant access to the dynamic user privilege level or specify access rights to the **configure** command to enable access to the configuration command mode.

---

### TACACS+ Server Sample Configuration

You can specify authorization for each command in the TACACS+ server database.

The following example shows how to configure authorization on a TACACS+ server for the user Zoe:

```
user=Zoe {
 cmd = protect {
 permit .*
 }
 cmd = "no protect" {
 permit .*
 }
 cmd = learning {
 deny policy*
 }
 cmd = "no learning" {
 deny .*
 }
 cmd = dynamic-filter {
 permit .*
 }
 cmd = "no dynamic-filter" {
 permit .*
 }
 cmd = flex-filter {
 deny .*
 }
 cmd = "no flex-filter" {
 deny .*
 }
}
```

### Disabling Tab Completion of Zone Names

You can limit the access to the zone configurations to authorized users only by disabling the tab completion feature when entering the zone names. This setting applies to all commands in which you specify the zone name.

When you enter commands in global or configuration mode, such as the **zone** command, **no zone** command, **show zone** command, and **deactivate** command, the Guard no longer displays or completes the zone name. You must enter the complete zone name to configure a zone, change the zone operation mode, or display zone statistics.

The Guard sends the **tab-complete zone-list** command to the TACACS+ server when you disable tab completion of zone names. Configure authorization for the **tab-complete zone-list** command on the TACACS+ server to enable tab completion of zone names to authorized users.

The following example shows how to disable tab completion of zone names for all the **zone** commands:

```
user@GUARD-conf# aaa authorization commands zone-completion tacacs+
```

To enable tab completion of zone names, use the **no** form of the command.

## Configuring Accounting

Accounting management allows you to track the services that users are accessing and save the accounting information on a TACACS+ server. You can enable accounting of requested services for billing, reporting, or security purposes. By default, the Guard is configured with accounting management disabled.

To configure accounting, perform the following steps:

- Step 1** Configure the TACACS+ server connection. See the “[Configuring the TACACS+ Server Attributes](#)” section on page 3-14 for more information.
- Step 2** Configure accounting by entering the following command in configuration mode:

```
aaa accounting commands {show | dynamic | config | admin} stop-only {local | tacacs+}
```

[Table 3-8](#) provides the keywords for the **aaa accounting** command.

**Table 3-8** Keywords for the **aaa accounting** Command

| Parameter                              | Description                                                                                                                    |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>show   dynamic   config   admin</b> | Defines accounting for the specified privilege level (see <a href="#">Table 2-1</a> for information on user privilege levels). |
| <b>stop-only</b>                       | Records the action when the command execution terminates.                                                                      |
| <b>tacacs+</b>                         | Uses a TACACS+ server database to record accounting information.                                                               |
| <b>local</b>                           | Does not save accounting information.                                                                                          |

To configure accounting for more than one privilege level, enter the **aaa accounting** command for each privilege level for which accounting is required.

We recommend that you enable accounting management for the config user privilege level only. Tracking and saving accounting data may affect Guard performance.

Use the **no** form of the command to remove the accounting management for a privilege level.

The following example shows how to configure accounting for commands that require the config privilege level on a TACACS+ server:

```
user@GUARD-conf# aaa accounting commands config stop-only tacacs+
```

## Configuring the TACACS+ Server Attributes

You must configure the TACACS+ server attributes to enable authentication, authorization, or accounting with a TACACS+ server.

**Caution**

You must configure the TACACS+ server attributes before you apply the TACACS+ authentication method or you may not be able to access the Guard.

To configure the TACACS+ server attributes, perform the following steps:

- 
- Step 1** Configure the IP address of the TACACS+ server by entering the **tacacs-server host** *ip-address* command.
- See the “[Configuring a TACACS+ Server IP Address](#)” section on page 3-15 for more information.
- Step 2** Configure the encryption key that the Guard uses to access the TACACS+ server by entering the **tacacs-server key** *tacacs-key* command.
- See the “[Configuring the TACACS+ Server Encryption Key](#)” section on page 3-15 for more information.
- Step 3** (Optional) Configure the search method that the Guard uses for authentications by entering the **tacacs-server first-hit** command.
- See the “[Configuring the TACACS+ Search Method](#)” section on page 3-16 for more information.
- Step 4** (Optional) Configure the TACACS+ server connection timeout by entering the **tacacs-server timeout** *timeout* command.
- See the “[Configuring the TACACS+ Server Connection Timeout](#)” section on page 3-16 for more information.
- Step 5** Display the TACACS+ server connection statistics by entering the **show tacacs statistics** command.
- See the “[Displaying TACACS+ Server Statistics](#)” section on page 3-16 for more information.
- 

The Guard user privilege levels relate to the TACACS+ privilege numeration as follows:

- **admin** = 15
- **config** = 10
- **dynamic** = 5
- **show** = 0

This section contains the following topics:

- [Configuring a TACACS+ Server IP Address](#)
- [Configuring the TACACS+ Server Encryption Key](#)
- [Configuring the TACACS+ Search Method](#)
- [Configuring the TACACS+ Server Connection Timeout](#)
- [Displaying TACACS+ Server Statistics](#)

## Configuring a TACACS+ Server IP Address

You can configure the Guard to use a sequential list of TACACS+ servers for authentication, authorization, and accounting. The Guard uses the TACACS+ server list to authenticate or authorize users or send an accounting event; if that server does not respond, the Guard selects the second server. Authentication or authorization fails only if all servers listed do not respond.

Alternatively, you can configure the Guard to use only the first TACACS+ server on the list to authenticate users (see the “[Configuring the TACACS+ Search Method](#)” section on page 3-16 for more information).

You must define the IP address of each TACACS+ server on the list. You can define a maximum of nine TACACS+ servers.

To add a TACACS+ server to the list and assign its IP address, use the following command in configuration mode:

```
tacacs-server host ip-address [port port_number]
```

The *ip-address* argument specifies the IP address of the TACACS+ server.

[Table 3-9](#) provides the arguments and keywords for the **tacacs-server host** command.

**Table 3-9 Arguments and Keywords for the tacacs-server host Command**

| Parameter                      | Description                                                                                                                      |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>              | IP address of the TACACS+ server. Enter the IP address in dotted-decimal notation (for example, an IP address of 192.168.100.1). |
| <b>port</b> <i>port_number</i> | (Optional) Specifies the port number to use. If you do not specify a port number, the Guard uses port 49 by default.             |

The TACACS+ servers are added to the list in the order in which you enter them. You can add a maximum of nine servers to the list.

The following example shows how to add a server to the TACACS+ server list:

```
user@GUARD-conf# tacacs-server host 192.168.33.45 port 60
```

## Configuring the TACACS+ Server Encryption Key

You must configure the encryption key to access a TACACS+ server. The key must match the key on the TACACS+ servers. The key cannot contain spaces.

To configure the server encryption access key, use the following command in configuration mode:

```
tacacs-server key tacacs-key
```

The argument *tacacs-key* is an alphanumeric string that contains up to 100 characters.



### Note

You can define only one encryption key. When using several TACACS+ servers, the Guard uses the same key to encrypt communication with all TACACS+ servers.

The following example shows how to set the TACACS+ server encryption key to MyKey:

```
user@GUARD-conf# tacacs-server key MyKey
```

## Configuring the TACACS+ Search Method

You can configure the Guard to consider an authentication rejection as final and stop further searching with other TACACS+ servers by using the **tacacs-server first-hit** command in configuration mode. The Guard performs user authentication using only the first TACACS+ server on the server list to respond. If the first TACACS+ server does not respond, the Guard selects the next server on the list. The Guard regards the first user authentication approval or rejection received as the final decision and stops attempting to authenticate the user with other TACACS+ servers.

To configure the Guard to continue a sequential search of the defined TACACS+ servers in an attempt to find a server that accepts the user authentication, use the **no tacacs-server first-hit** command in configuration mode. This method is the default setting for the first-hit operation. User authentication fails if all of the defined TACACS+ servers reject the user authentication or the Guard cannot communicate with any of the servers.

The following example shows how to configure the TACACS+ search method so that the Guard uses only the first TACACS+ server on the list to authenticate users:

```
user@GUARD-conf# tacacs-server first-hit
```

## Configuring the TACACS+ Server Connection Timeout

You can configure the amount of time that the Guard waits for a reply from the TACACS+ server. When the timeout ends, the Guard either attempts to establish a connection with the next TACACS+ server (if a server was configured) or falls back to local AAA (if a fallback was configured). Authentication and authorization fail if no fallback method is configured.



### Note

---

The same server timeout is used for communication with all TACACS+ servers.

---

To configure the TACACS+ server connection timeout, use the following command in configuration mode:

```
tacacs-server timeout timeout
```

The *timeout* argument specifies the amount of time (in seconds) that the Guard waits for a TACACS+ server to reply. The default timeout is 0.

The following example shows how to configure the TACACS+ server connection timeout to 600 seconds:

```
user@GUARD-conf# tacacs-server timeout 600
```



### Tip

---

You may want to increase the timeout value if you have network problems or if the TACACS+ servers are slow to respond and cause persistent timeouts.

---

## Displaying TACACS+ Server Statistics

You can display statistical information for the TACACS+ servers that you define by using the **show tacacs statistics** command in configuration mode.

To clear the TACACS+ statistics, use the **clear tacacs statistics** command in configuration mode.

Table 3-10 displays the fields in the `show tacacs statistics` command output.

**Table 3-10** Field Descriptions in the `show tacacs statistics` Command Output

| Field | Description                                                                                                   |
|-------|---------------------------------------------------------------------------------------------------------------|
| PASS  | Specifies the number of times that the Guard accessed the TACACS+ server successfully and was granted access. |
| FAIL  | Specifies the number of times that the Guard accessed the TACACS+ server successfully and was denied access.  |
| ERROR | Specifies the number of times that the Guard could not access the TACACS+ server.                             |

## Establishing Communication with the Detector

You can establish a secure communication channel between a Detector and the Guards that you define on the Detector remote Guard lists. The secure communication channel enables the Detector to perform the following tasks:

- **Activate the Guard**—When the Detector detects a zone traffic anomaly, it uses the communication channel to activate the Guard that provides zone protection and to poll the Guard during zone protection.
- **Synchronize a zone configuration**—The Detector uses the communication channel to exchange zone configuration information with the Guard.

After you configure the communication channel parameters on both the Detector and Guard, from the Detector you initiate a connection with the Guard which enables the Detector to exchange the keys and certificates that are required to establish a secure communication channel with the Guard. The Detector then closes the connection and establishes the communication channel when it needs to activate the Guard, synchronize a zone configuration, or poll the Guard.

The Detector and Guard support the following two types of communication channels:

- **Secure Shell (SSH) version 2**—Enables the Detector to activate the Guard.
- **Secure Sockets Layer (SSL)**—Enables the Detector to activate the Guard, poll the Guard, and synchronize zone configurations.

You use the zone remote Guard list and default remote Guard list on the Detector to specify the Guards that the Detector communicates with for zone protection and synchronization. When you specify a Guard on a remote Guard list, you select the type of communication channel that the Detector is to establish with the Guard: SSH or SSL. Both devices require the SSH service for establishing a SSH or SSL communication channel. By default, the SSH service is always enabled on both devices. When you establish an SSL communication channel, the Detector uses the SSH communication channel only for the initial connection with a Guard, during which time the devices exchange their keys and certificates.



### Note

Before you can establish a communication channel between a Detector and a Guard, you must add the Guard to a remote Guard list on the Detector. For more information, see the *Cisco Traffic Anomaly Detector Module Configuration Guide* and *Cisco Traffic Anomaly Detector Configuration Guide*.

This section contains the following topics:

- [Configuring the SSL Communication Channel Parameters](#)
- [Configuring the SSH Communication Channel Parameters](#)

## Configuring the SSL Communication Channel Parameters

Configure an SSL communication channel between the Detector and the Guard when you need the Detector to interact with the Guard as follows:

- Activate the Guard when the Detector detects a traffic anomaly.
- Synchronize zone configurations with the Guard.
- Poll the Guard to identify that an attack on the zone has ended. If you enable the detect and learn process on the Detector, the Detector suspends the learning process (threshold tuning) when it detects an attack on the zone. The Detector polls the Guard that it activated to mitigate the attack to determine when the attack is over, at which point, the Detector automatically resumes the learning process.
- Monitor communication with the Guard and notify you if remote actions fail, such as activating the Guard to protect the zone.

An SSL communication channel provides secure connections through a combination of authentication and data encryption and relies upon digital certificates, private-public key exchange pairs, and Diffie-Hellman key agreement parameters for this level of security. SSL encrypts the data so that only the intended recipient can decipher the data.

Each Guard and Detector uses a digital certificate to authenticate the device attempting to communicate with it over the communication channel. The identity of the Guard and the Detector in the SSL certificates is associated with the device IP address. To ensure a secure connection, the Detector generates a private-public key pair and distributes its public key to the Guards that you define on the remote Guard lists.

After you configure the SSL communication parameters on both the Guard and the Detector, you must establish the communication channel between the two devices, which you perform from the Detector. During the initial connection to the Guard, the Detector establishes an SSH communication channel with the user riverhead on the Guard and then the devices exchange the keys and certificates that are required to secure the communication channel. After the initial connection, the Detector establishes an SSL communication channel when needed to activate the Guard, poll the Guard, or synchronize zone configurations.

If you replace one of the devices on either end of an SSL communication channel or change one of their IP addresses, then you must regenerate the SSL certificates in both devices so that the two devices can successfully authenticate each other.

This section contains the following topics:

- [Enabling an SSL Communication Channel](#)
- [Regenerating SSL Certificates](#)

### Enabling an SSL Communication Channel

To enable an SSL communication channel, you must configure the Guard and the Detector to allow the following connection types:

- SSH—The Detector establishes the initial connection with the Guard using an SSH communication channel to exchange the keys and certificates.
- SSL—The Detector uses an SSL communication channel to establish all connections with the Guard after the initial connection.

**Caution**

If the Guard is authenticating users using TACACS+ authentication, you must define the user riverhead on the TACACS+ server to enable the Detector to establish the SSH communication channel during the initial connection with the Guard.

To enable an SSL communication channel, perform the following steps on both the Guard and the Detector:

- 
- Step 1** Permit access to the SSH service by the companion device IP address by entering the **permit ssh ip-address-general [ip-mask]** in configuration mode.
- The *ip-address-general* and *ip-mask* arguments define the IP address of the companion device.
- Step 2** Enable the communication channel service by entering the **service internode-comm** command in configuration mode.
- Step 3** Permit access to the communication channel service by the companion device IP address by entering the **permit internode-comm ip-address-general [ip-mask]** command in configuration mode.
- The *ip-address-general* and *ip-mask* arguments define the IP address of the companion device.
- 

After you configure the Guard and the Detector to enable the SSL communication channel, you can establish the communication channel between them. For information on establishing the communication channel, see the *Cisco Traffic Anomaly Detector Module Configuration Guide* or the *Cisco Traffic Anomaly Detector Configuration Guide*.

## Regenerating SSL Certificates

The key that identifies the Guard and the Detector in the SSL certificates is associated with the device IP addresses. You must regenerate new SSL certificates for the Guard and the Detector on both ends of a communication channel when you make the following changes:

- Change the IP address of one of the devices.
- Replace one of the devices.

The process of regenerating new SSL certificates includes deleting the current certificates from both devices. To display the current SSL certificates, use the **show internode-comm certs** command.

To regenerate the current SSL certificates, perform the following steps:

- 
- Step 1** From the Detector, delete the SSL certificate of the Guard by using the following command in configuration mode:

```
cert remove cert-host-ip
```

The *cert-host-ip* argument specifies the IP address of the Guard. Enter an asterisk (\*) to delete the SSL certificates of all Guards.

The following example shows how to delete an SSL certificate:

```
user@DETECTOR-conf# cert remove 10.56.36.4
```

- Step 2** From the Guard, delete the SSL certificate of the Detector by using the following command in configuration mode:

```
cert remove cert-host-ip
```

The *cert-host-ip* argument specifies the IP address of the Detector. Enter an asterisk (\*) to delete the SSL certificates of all Detectors that have established communication channels with the Guard.

- Step 3** If you replace the Guard, then you must also delete its SSH host key from the Detector. From the Detector, use the following command in configuration mode to delete Guard SSH host keys:

```
no host-keys ip-address-general
```

The *ip-address-general* argument specifies the IP address of the remote device.

- Step 4** From the Detector, regenerate new SSL certificates by establishing a new SSL communication channel between the Guard and the Detector. For information about establishing a communication channel, see the *Cisco Traffic Anomaly Detector Module Configuration Guide* or the *Cisco Traffic Anomaly Detector Configuration Guide*.
- 

## Configuring the SSH Communication Channel Parameters

Configure an SSH communication channel between the Detector and the Guard when the only interaction between the two devices that you need is for the Detector to activate the Guard when it detects a traffic anomaly. An SSH communication channel does not allow the Detector to perform the following tasks with the Guard:

- Synchronize zone configurations with the Guard.
- Poll the Guard to identify that an attack on the zone has ended. If you enable the detect and learn process on the Detector, the Detector suspends the learning process (threshold tuning) when it detects an attack on the zone. Because the Detector cannot poll the Guard to determine when the attack is over, it is unable to automatically resume the learning process when the attack ends.
- Monitor communication with the Guard and notify you if remote actions fail, such as activating the Guard to protect the zone.

To allow the Detector to perform these tasks, you must configure an SSL communication channel (see the [“Configuring the SSL Communication Channel Parameters”](#) section on page 3-18).

To ensure a secure SSH communication channel, the Detector generates a private-public SSH key pair and distributes the public SSH key to the Guards listed in the remote Guard lists.

After you enable the SSH communication channel, you must establish the communication channel between the Detector and the Guard, which you perform from the Detector.

If you replace a Guard at one end of an SSH communication channel, then you must regenerate the SSH private (host) and public keys on the Detector so that it can successfully authenticate itself with the new Guard.

This section contains the following topics:

- [Enabling an SSH Communication Channel](#)
- [Regenerating SSH Communication Channel Keys](#)

## Enabling an SSH Communication Channel

To enable an SSH communication channel between a Guard and a Detector, permit access to the SSH service on the Guard from the Detector IP address by entering the **permit ssh** command in configuration mode



### Caution

If the Guard is authenticating users using TACACS+ authentication, you must define the user riverhead on the TACACS+ server to enable the Detector to establish the SSH communication channel.

After you enable the SSH communication channel between the Guard and the Detector, you can establish the communication channel between them. For information on establishing the communication channel, see the *Cisco Traffic Anomaly Detector Module Configuration Guide* or the *Cisco Traffic Anomaly Detector Configuration Guide*.

## Regenerating SSH Communication Channel Keys

If you replace a Guard that a Detector communicates with over an SSH communication channel, then you must perform the following steps to regenerate the SSH communication channel keys:

- 
- Step 1** Delete the SSH host key from the Detector by entering the **no host-keys ip-address-general** configuration mode command on the Detector.
- The *ip-address-general* argument specifies the IP address of the remote device.
- To display the host keys listed on the Guard, use the **show host-keys** command.
- Step 2** Configure the SSH key on the remote Guard by performing one of the following actions:
- Establish a new SSH communication channel from the Detector.
  - Add the Detector public key manually to the remote Guard. You can copy the Detector public SSH key, and paste it into the list of SSH keys that the Guard maintains.
- To display the Detector public SSH key, use the **show public-key** command on the Detector.
- To add the Detector public SSH key to the list of SSH keys that the Guard maintains, use the **key add** command on the Guard. See the [“Adding SSH Keys” section on page 3-23](#) for more information.
- 

## Configuring a Date and a Time

You can set the time and the date by using the following command in configuration mode:

```
date MMDDhhmm[[CC]YY][.ss]
```

Table 3-11 provides the arguments for the **date** command.

**Table 3-11 Arguments for the date Command**

| Parameter  | Description                                                          |
|------------|----------------------------------------------------------------------|
| <i>MM</i>  | Month in numeric figures.                                            |
| <i>DD</i>  | Day of the month.                                                    |
| <i>hh</i>  | Hour in a 24-hour clock.                                             |
| <i>mm</i>  | Minutes.                                                             |
| <i>CC</i>  | (Optional) First two digits of the year (for example, <b>2005</b> ). |
| <i>YY</i>  | (Optional) Last two digits of the year (for example, <b>2005</b> ).  |
| <i>.ss</i> | (Optional) Seconds (the decimal point must be present).              |

The following example shows how to configure the date to October 8 of the year 2007 and the time to 5:10 pm (1710) and 17 seconds:

```
user@GUARD-conf# date 1008171003.17
Wed Oct 8 17:10:17 EDT 2003
```

## Synchronizing the Guard Clock with an NTP Server

To configure the Guard system clock to synchronize with a Network Time Protocol (NTP) server, perform the following steps in configuration mode:

---

**Step 1** Configure the date and time locally by entering the following command:

```
date MMDDhhmm[[CC] YY] [.ss]
```

See the “[Configuring a Date and a Time](#)” section on page 3-21 for more information.

**Step 2** Configure the Guard system time zone by entering the following command:

```
timezone timezone-name
```

The *timezone-name* argument specifies the name of the time zone. The name is composed of the *continent* / *city* options.

The following are the continent options:

- Africa, America, Antarctica, Arctic, Asia, Atlantic, Australia, Europe, Indian, Pacific
- Etc—Wildcard for a desired time zone




---

**Tip** The time zone name is case sensitive. Type the desired continent name and press **TAB** twice for a list of relevant cities.

---

**Step 3** Enable the NTP service by entering the following command:

```
service ntp
```

**Step 4** Permit access to the NTP service from a network address by entering the following command:

```
permit ntp ip-address
```

**Step 5** Configure the IP address of the desired NTP server by entering the following command:

```
ntp server ip-address
```

The *ip-address* argument specifies the NTP server IP address.

**Step 6** Reload the Guard configuration.

The following example shows how to configure an NTP server:

```
user@GUARD-conf# date 1008171003.17
user@GUARD-conf# timezone Africa/Timbuktu
user@GUARD-conf# service ntp
user@GUARD-conf# permit ntp 192.165.200.224
user@GUARD-conf# ntp server 192.165.200.224
```

## Managing SSH Keys

The Guard supports SSH for secure remote login. You can add a list of SSH keys to enable secure communication from a remote device to the Guard without entering a login and password.

The section describes how you can manage the Guard SSH key list and contains the following topics:

- [Adding SSH Keys](#)
- [Deleting SSH Keys](#)

### Adding SSH Keys

You can enable an SSH connection without entering a login and password by adding the remote connection SSH public key to the Guard SSH key list.

Enter the following command in configuration mode:

```
key add [user-name] {ssh-dsa | ssh-rsa} key-string comment
```

[Table 3-12](#) provides the arguments and keywords for the **key add** command.

**Table 3-12 Arguments and Keywords for the key add Command**

| Parameter        | Description                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>user-name</i> | (Optional) Name of the user to which the SSH key is added. Only an administrator can add an SSH key for other users.<br>The default is to add the SSH key for the current user. |
| <b>ssh-dsa</b>   | Specifies the SSH2-DSA key type.                                                                                                                                                |
| <b>ssh-rsa</b>   | Specifies the SSH2-RSA key type.                                                                                                                                                |

**Table 3-12 Arguments and Keywords for the key add Command (continued)**

| Parameter         | Description                                                                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>key-string</i> | Public SSH key that was created on a Cisco Traffic Anomaly Detector or remote terminal. The key string is limited to 8192 bits.<br><br>You must copy the complete key and exclude the key type identification (ssh-rsa or ssh-dsa).                                   |
| <i>comment</i>    | Device description. The comment format is usually in the format of user@hostname for the user and machine used to generate the key. For example, the default comment used for the SSH public keys that the Cisco Traffic Anomaly Detector generates is root@DETECTOR. |

The following example shows how to add an SSH RSA key:

```
user@GUARD-conf# key add ssh-rsa 14513797528175730. . .user@Guard.com
```

## Deleting SSH Keys

You can remove an SSH key from the list. If you remove the SSH key, you must authenticate the next time that you establish an SSH session with the Guard.

To remove an SSH key from the Guard, use the following command in configuration mode:

```
key remove [user-name] key-string
```

Table 3-13 provides the arguments for the **key remove** command.

**Table 3-13 Arguments for the key remove Command**

| Parameter         | Description                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>user-name</i>  | (Optional) Name of the user from which the SSH keys are removed.<br><br>Only an administrator can delete an SSH key for other users. The default is to delete the SSH key for the current user. |
| <i>key-string</i> | Public SSH key to delete.<br><br>Paste the SSH public key onto the prompt. Paste only the key without the identification field (ssh-rsa or ssh-dsa).                                            |

The following example shows how to view a user key so that it can be cut and pasted into the **key remove** command:

```
user@GUARD-conf# show keys Lilac
ssh-rsa 2352345234523456... user@Guard.com
user@GUARD-conf# key remove Lilac 2352345234523456...
```

# Configuring the Keys for SFTP and SCP Connections

Secure File Transfer Protocol (SFTP), which is layered on top of SSH2, and Secure Copy Protocol (SCP), which relies on SSH, provide a secure and authenticated method for copying files. SFTP and SCP use public key authentication and strong data encryption, which prevents login, data, and session information from being intercepted or modified in transit.

To configure the keys for SFTP and SCP connections, perform the following steps:

- Step 1** Display the Guard public key on the Guard by entering the **show public-key** command in configuration mode.
- If the key exists, skip [Step 2](#) and proceed to [Step 3](#).
- If no key exists, proceed to [Step 2](#).
- Step 2** Generate a private-public key pair on the Guard by entering the **key generate** command in configuration mode.
- If an SSH key pair already exists, the following message appears:
- ```
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)?
```
- Type **y** to regenerate the key.
- The Guard creates the public-private key pair. To display the Guard public key, use the **show public-key** command in configuration mode.
- Step 3** Copy the public key from the Guard and paste it into the key file on the network server.
- For example, if you are connecting to a network server that is installed on a Linux operating system with the username `user` account, add the Guard public key to the `/home/username/.ssh/authorized_keys2` file.
- Make sure that the key is copied as a single line. If the key is copied as two lines, delete the new line character at the end of the first line.

**Note**

If you do not copy the public key and paste it into the key file on the network server, you cannot configure automatic export functions (such as the **export reports** command) and you have to enter your password each time that you manually connect to the network server.

Changing the Hostname

You can change the hostname of the Guard. The change takes effect immediately, and the new hostname is automatically integrated into the CLI prompt string.

To change the Guard hostname, use the following command in configuration mode:

```
hostname name
```

The *name* argument specifies the new hostname.

The following example shows how to change the hostname of the Guard:

```
user@GUARD-conf# hostname CiscoGuard  
admin@CiscoGuard-conf#
```

Enabling SNMP Traps

You can enable the Guard to send SNMP traps and notify you of significant events that occur on the Guard. In addition, you can configure the Guard SNMP trap generator parameters and define the scope of the SNMP trap information that the Guard reports.

A trap is logged in the Guard event log and displayed in the event monitor when a trap condition occurs, regardless of whether the SNMP agent sends the trap.

To configure the Guard to send SNMP traps, perform the following steps:

-
- Step 1** Enable the SNMP trap generator service by entering the following command in configuration mode:
- ```
service snmp-trap
```
- Step 2** Configure the SNMP trap generator parameters (the trap destination IP address and the trap information scope) by entering the following command:
- ```
snmp trap-dest ip-address [community-string [min-severity]]
```

Table 3-14 provides the arguments for the **snmp trap-dest** command.

Table 3-14 Arguments for the snmp trap-dest Command

| Parameter | Description |
|-------------------------|--|
| <i>ip-address</i> | Destination host IP address. |
| <i>community-string</i> | (Optional) Community string that is sent with the trap. This string must match the community string defined for the destination host. The default community string is <i>public</i> . Enter an alphanumeric string from 1 to 15 characters. The string cannot contain spaces. |
| <i>min-severity</i> | (Optional) Trap information scope. Define the scope by stating the minimum severity-level coverage. The trap then displays all specified severity-level events and above. For example, if you specify Warnings, the trap displays all severity-level events from Warnings to Emergencies. The following list states the severity-level options: <ul style="list-style-type: none"> • Emergencies—System is unusable (severity=0). • Alerts—Immediate action needed (severity=1). • Critical—Critical conditions (severity=2). • Errors—Error conditions (severity=3). • Warnings—Warning conditions (severity=4). • Notifications—Normal but significant conditions (severity=5). • Informational—Informational messages (severity=6). • Debugging—Debugging messages (severity=7). By default, the report displays all severity-level events. |

To delete SNMP trap generator parameters, use the **no snmp trap-dest** command. Enter an asterisk (*) to remove all SNMP trap destination parameters.

The following example shows that traps with a severity level equal to or higher than errors are sent to the destination IP address 192.168.100.52 with the SNMP community string of tempo:

```
user@GUARD-conf# snmp trap-dest 192.168.100.52 tempo errors
```

Table 3-15 lists the SNMP traps that the Guard generates.

Table 3-15 *SNMP Traps*

| SNMP Trap | Severity | Description |
|----------------------------|-----------|--|
| rhExcessiveUtilizationTrap | EMERGENCY | The Guard cannot add new dynamic filters because more than 150,000 dynamic filters are active concurrently in all the Guard zones. |
| rhExcessiveUtilizationTrap | EMERGENCY | The anomaly detection engine memory limit was reached (higher than 90 percent). |
| rhGeneralTrap | EMERGENCY | The Guard failed to activate zone protection by using the packet activation method and will send subsequent traps with an ALERT severity level. |
| rhGeneralTrap | EMERGENCY | The Guard failed to activate diversion. |
| rhGeneralTrap | EMERGENCY | The Guard restored diversion. |
| rhGeneralTrap | ALERT | The Guard failed to activate zone protection by using the packet activation method, and a trap with an EMERGENCY severity level was already generated. |
| rhGeneralTrap | ALERT | The disk space is 80 percent. |
| rhExcessiveUtilizationTrap | CRITICAL | The Gigabit interface link utilization in bps ¹ is above 85 percent. |
| rhExcessiveUtilizationTrap | CRITICAL | The memory utilization is above 85 percent. |
| rhExcessiveUtilizationTrap | CRITICAL | The accelerator card CPU utilization is above 85 percent. |
| rhGeneralTrap | CRITICAL | The HW diagnostics card reported an error. |
| rhLinkStatusTrap | CRITICAL | The link is down. |
| rhDynamicFilterTrap | ERROR | The number of pending dynamic filters is 1000, and new pending dynamic filters will be discarded. |
| rhZoneGenericTrap | ERROR | The Guard failed to synchronize the zone configuration. |

Table 3-15 SNMP Traps (continued)

| SNMP Trap | Severity | Description |
|----------------------------|---------------|---|
| rhGeneralTrap | ERROR | The Guard failed to activate zone protection as follows: <ul style="list-style-type: none"> From protect or from learn to protect and learn From protect and learn to protect or to learn The Guard deactivated zone protection and the learning process. |
| rhDynamicFilterTrap | WARNING | The Guard failed to add dynamic filters. |
| rhExcessiveUtilizationTrap | WARNING | The Guard has more than 135,000 dynamic filters that are active concurrently in all the zones. When the number of active dynamic filters reaches 150,000, the Guard cannot add new dynamic filters. |
| rhGeneralTrap | WARNING | The disk space is 75 percent. |
| rhPolicyConstructionTrap | WARNING | The policy construction phase of the learning process has failed. |
| rhProtectionTrap | WARNING | The Guard failed to start zone protection. |
| rhReloadTrap | WARNING | The Guard has restarted. The trap contains a MIB2 warm-start or cold-start trap and information on what caused the Guard to restart. |
| rhReloadTrap | WARNING | The Guard has shut down. The trap contains a MIB2 warm-start or cold-start trap and information on what caused the Guard to shut down. |
| rhThresholdTuningTrap | WARNING | The threshold tuning phase of the learning process has failed. |
| rhAttackTrap | NOTIFICATIONS | An attack has started. |
| rhAttackTrap | NOTIFICATIONS | An attack has ended. |
| rhLinkStatusTrap | NOTIFICATIONS | The link is up. |
| rhPolicyConstructionTrap | NOTIFICATIONS | The policy construction phase of the learning process has been started. |
| rhPolicyConstructionTrap | NOTIFICATIONS | The policy construction phase of the learning process has been accepted. |
| rhPolicyConstructionTrap | NOTIFICATIONS | The policy construction phase of the learning process has been stopped. |
| rhProtectionTrap | NOTIFICATIONS | The zone protection has started. |
| rhProtectionTrap | NOTIFICATIONS | The zone protection has ended. |
| rhThresholdTuningTrap | NOTIFICATIONS | The threshold tuning phase of the learning process has been started. |
| rhThresholdTuningTrap | NOTIFICATIONS | The threshold tuning phase of the learning process has been accepted. |

Table 3-15 SNMP Traps (continued)

| SNMP Trap | Severity | Description |
|----------------------------|---------------|--|
| rhThresholdTuningTrap | NOTIFICATIONS | The threshold tuning phase of the learning process has been stopped. |
| rhZoneGenericTrap | NOTIFICATIONS | The Guard has started to synchronize the zone configuration. |
| rhZoneTrap | NOTIFICATIONS | A new zone has been created. |
| rhZoneTrap | NOTIFICATIONS | A zone has been deleted. |
| rhDynamicFilterControlTrap | INFO | The number of attack-detection events that the Guard did not send for a specific policy. |
| rhDynamicFilterControlTrap | INFO | The Guard has more than 1000 active dynamic filters and will not send traps for dynamic filters that it deletes. |
| rhDynamicFilterTrap | INFO | A dynamic filter has been added. |
| rhDynamicFilterTrap | INFO | A dynamic filter has been deleted. |
| rhDynamicFilterTrap | INFO | A pending dynamic filter has been added. |

1. bps = bits per second

Configuring SNMP Community Strings

You can access the Guard SNMP server and retrieve information as defined by the Management Information Base 2 (MIB2) and the Cisco Riverhead proprietary MIB. The community string acts like a password and permits read access from the Guard SNMP agent. You can configure the Guard SNMP community string and enable access to the SNMP agent from clients in different organizational units and with different community strings.

To add an SNMP community string, use the following command in configuration mode:

```
snmp community community-string
```

The *community-string* argument specifies the desired Guard community string. Enter an alphanumeric string from 1 to 15 characters. The string cannot contain spaces. The Guard default community string is riverhead. You can specify as many community names as you want. To delete a community string, use the **no community string** command. Enter an asterisk (*) to remove all SNMP community strings.

The following example shows how to configure the SNMP community string:

```
user@GUARD-conf# snmp community tempo
```

Configuring the Login Banner

The login banner is the text that appears on the screen before user authentication when you open an SSH session, a console port connection, or a WBM session to the Guard.

You can configure a login banner to warn users against unauthorized access, describe what is considered the proper use of the system, and to alert users that the system is being monitored to detect improper use and other illicit activity.

The Guard displays the login banner in the following locations:

- CLI—Before the password login prompt or as a popup window (depending on the SSH client that you are using).
- WBM—On the right side of the Guard login window.

This section contains the following topics:

- [Configuring the Login Banner from the CLI](#)
- [Importing the Login Banner](#)
- [Deleting the Login Banner](#)

Configuring the Login Banner from the CLI

You can create a single or multiple message banner by using the **login-banner** command. If you enter more than one login banner, the new login banner is appended to the existing login banner as a new line.

To configure the login banner, use the following command in configuration mode:

```
login-banner banner-str
```

The *banner-str* argument specifies the banner text. The maximum string length is 999 characters. If you use spaces in the expression, enclose the expression in quotation marks (“ ”).

To display the login banner, use the **show login-banner** command.

The following example shows how to configure and display the login banner:

```
user@GUARD-conf# login-banner "Welcome to the Cisco Guard"
user@GUARD-conf# login-banner "Unauthorized access is prohibited."
user@GUARD-conf# login-banner "Contact sysadmin@corp.com for access."
user@GUARD-conf# show login banner
Welcome to the Cisco Guard
Unauthorized access is prohibited.
Contact sysadmin@corp.com for access.
```

Importing the Login Banner

You can import a text file from a network server to replace the existing login banner by entering one of the following commands in global mode or in configuration mode:

- **copy ftp login-banner** *server full-file-name* [*login* [*password*]]
- **copy {sftp | scp} login-banner** *server full-file-name login*

The maximum length of each line in the file that you import is 999 characters.

Because SFTP and SCP rely on SSH for secure communication, if you do not configure the key that the Guard uses before you enter the **copy** command with the **sftp** or **scp** option, the Guard prompts you for the password. See the “[Configuring the Keys for SFTP and SCP Connections](#)” section on page 3-25 for more information on how to configure the key that the Guard uses for secure communication.

Table 3-16 provides the arguments and keywords for the **copy login-banner** command.

Table 3-16 Arguments and Keywords for the copy login-banner Command

| Parameter | Description |
|-----------------------|--|
| ftp | Specifies FTP. |
| sftp | Specifies SFTP. |
| scp | Specifies SCP. |
| <i>server</i> | IP address of the network server. Enter the IP address in dotted-decimal notation (for example, enter 192.168.10.2). |
| <i>full-file-name</i> | Complete name of the file. If you do not specify a path, the server copies the file from your home directory. |
| <i>login</i> | (Optional) The server login name. The <i>login</i> argument is optional when you define an FTP server. When you do not enter a login name, the FTP server assumes an anonymous login and does not prompt you for a password. |
| <i>password</i> | (Optional) Password for the remote FTP server. If you do not enter the password, the Guard prompts you for it. |

The following example shows how to import the login banner from an FTP server:

```
user@GUARD-conf# copy ftp login-banner 10.0.0.191 /root/login-banner <user> <password>
```

Deleting the Login Banner

If you no longer want to display a message before user authentication, delete the login banner.

To delete the login banner, use the **no login-banner** command in configuration mode.

The following example shows how to delete the login banner:

```
user@GUARD-conf# no login-banner
```

Configuring the WBM Logo

To customize your end-user interface, you can add a company logo or any customized logo to the WBM web pages.

The new logo appears in the following places:

- On the Guard login page, under the Cisco Systems logo.
- On all WBM pages, except for the Guard login page, on the right side of the Cisco Systems logo.

The new logo must be in GIF format. We recommend that the size of the new logo is as follows: width = 87 pixels and height = 41 pixels.

This section contains the following topics:

- [Importing the WBM Logo](#)
- [Deleting the WBM Logo](#)

Importing the WBM Logo

To import a new logo from a network server, use the following command in global mode or in configuration mode:

- **copy ftp wbm-logo** *server full-file-name* [*login* [*password*]]
- **copy {sftp | scp} wbm-logo** *server full-file-name login*

Because SFTP and SCP rely on SSH for secure communication, if you do not configure the key that the Guard uses before you enter the **copy** command with the **sftp** or **scp** option, the Guard prompts you for a password. See the “[Configuring the Keys for SFTP and SCP Connections](#)” section on page 3-25 for more information on how to configure the key that the Guard uses for secure communication.

Table 3-17 provides the arguments and keywords for the **copy wbm-logo** command.

Table 3-17 Arguments and Keywords for the copy wbm-logo Command

| Parameter | Description |
|-----------------------|--|
| ftp | Specifies FTP. |
| sftp | Specifies SFTP. |
| scp | Specifies SCP. |
| <i>server</i> | IP address of the network server. Enter the IP address in dotted-decimal notation (for example, enter 192.168.10.2). |
| <i>full-file-name</i> | Complete name of the file including the GIF file extension. If you do not specify a path, the server copies the file from your home directory. |
| <i>login</i> | (Optional) Server login name. The <i>login</i> argument is optional when you define an FTP server. When you do not enter a login name, the FTP server assumes an anonymous login and does not prompt you for a password. |
| <i>password</i> | (Optional) Password for the remote FTP server. If you do not enter the password, the Guard prompts you for it. |

The following example shows how to import the WBM logo file from an FTP server:

```
user@GUARD-conf# copy ftp wbm-logo 10.0.0.191 /root/WBMlogo.gif <user> <password>
```

Deleting the WBM Logo

To delete the WBM logo, use the **no wbm-logo** command in configuration mode.

The following example shows how to delete the login banner:

```
user@GUARD-conf# no wbm-logo
```

Configuring the Session Timeout

The session timeout is the amount of time that a session remains active when there is no activity. If there is no activity for the configured time, a timeout occurs, and then you must log in again. The session timeout is disabled by default.

The session timeout applies to the CLI only and does not apply to the WBM.

You can configure the number of minutes until the Guard disconnects an idle session automatically by entering the following command in configuration mode:

```
session-timeout timeout-val
```

The *timeout-val* argument specifies the number of minutes until the Guard disconnects an idle session automatically. Valid values are from 1 to 1440 minutes (one day).

The following example shows how to configure the Guard to disconnect an idle session after 10 minutes:

```
user@GUARD-conf# session-timeout 10
```

To prevent the Guard from disconnecting idle sessions automatically, use the **no session-timeout** command.

To display the value of the session timeout, use the **show session-timeout** command.

