



# CHAPTER 40

## Managing System Access

---

This chapter describes how to access the security appliance for system management through Telnet, SSH, and HTTPS. It also describes how to authenticate and authorize users and how to create login banners.

This chapter includes the following sections:

- [Allowing Telnet Access, page 40-1](#)
- [Allowing SSH Access, page 40-2](#)
- [Allowing HTTPS Access for ASDM, page 40-3](#)
- [Configuring ASDM and WebVPN on the Same Interface, page 40-4](#)
- [Configuring AAA for System Administrators, page 40-5](#)
- [Configuring a Login Banner, page 40-16](#)



### Note

---

To access the security appliance interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to the sections in this chapter.

---

## Allowing Telnet Access

The security appliance allows Telnet connections to the security appliance for management purposes. You cannot use Telnet to the lowest security interface unless you use Telnet inside an IPSec tunnel.

The security appliance allows a maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided between all contexts.

To configure Telnet access to the security appliance, follow these steps:

- 
- Step 1** To identify the IP addresses from which the security appliance accepts connections, enter the following command for each address or subnet:

```
hostname(config)# telnet source_IP_address mask source_interface
```

If there is only one interface, you can configure Telnet to access that interface as long as the interface has a security level of 100.

- Step 2** (Optional) To set the duration for how long a Telnet session can be idle before the security appliance disconnects the session, enter the following command:

```
hostname(config)# telnet timeout minutes
```

Set the timeout from 1 to 1440 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to let a host on the inside interface with an address of 192.168.1.2 access the security appliance, enter the following command:

```
hostname(config)# telnet 192.168.1.2 255.255.255.255 inside
hostname(config)# telnet timeout 30
```

To allow all users on the 192.168.3.0 network to access the security appliance on the inside interface, enter the following command:

```
hostname(config)# telnet 192.168.3.0 255.255.255.0 inside
```

## Allowing SSH Access

The security appliance allows SSH connections to the security appliance for management purposes. The security appliance allows a maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided between all contexts.

SSH is an application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities. The security appliance supports the SSH remote shell functionality provided in SSH Versions 1 and 2 and supports DES and 3DES ciphers.



### Note

XML management over SSL and SSH are not supported.

This section includes the following topics:

- [Configuring SSH Access, page 40-2](#)
- [Using an SSH Client, page 40-3](#)

## Configuring SSH Access

To configure SSH access to the security appliance, follow these steps:

**Step 1** To generate an RSA key pair, which is required for SSH, enter the following command:

```
hostname(config)# crypto key generate rsa modulus modulus_size
```

The modulus (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a value of 1024.

**Step 2** To save the RSA keys to persistent Flash memory, enter the following command:

```
hostname(config)# write mem
```

**Step 3** To identify the IP addresses from which the security appliance accepts connections, enter the following command for each address or subnet:

```
hostname(config)# ssh source_IP_address mask source_interface
```

The security appliance accepts SSH connections from all interfaces, including the one with the lowest security level.

- Step 4** (Optional) To set the duration for how long an SSH session can be idle before the security appliance disconnects the session, enter the following command:

```
hostname(config)# ssh timeout minutes
```

Set the timeout from 1 to 60 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

---

For example, to generate RSA keys and let a host on the inside interface with an address of 192.168.1.2 access the security appliance, enter the following command:

```
hostname(config)# crypto key generate rsa modulus 1024  
hostname(config)# write mem  
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside  
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside  
hostname(config)# ssh timeout 30
```

To allow all users on the 192.168.3.0 network to access the security appliance on the inside interface, the following command:

```
hostname(config)# ssh 192.168.3.0 255.255.255.0 inside
```

By default SSH allows both version one and version two. To specify the version number enter the following command:

```
hostname(config)# ssh version version_number
```

The *version\_number* can be 1 or 2.

## Using an SSH Client

To gain access to the security appliance console using SSH, at the SSH client enter the username **pix** and enter the login password set by the **password** command (see the [“Changing the Login Password”](#) section on page 8-1).

When starting an SSH session, a dot (.) displays on the security appliance console before the SSH user authentication prompt appears, as follows:

```
hostname(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears at the console when generating a server key or decrypting a message using private keys during SSH key exchange before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the security appliance is busy and has not hung.

## Allowing HTTPS Access for ASDM

To use ASDM, you need to enable the HTTPS server, and allow HTTPS connections to the security appliance. All of these tasks are completed if you use the **setup** command. This section describes how to manually configure ASDM access.

The security appliance allows a maximum of 5 concurrent ASDM instances per context, if available, with a maximum of 32 ASDM instances between all contexts.

To configure ASDM access, follow these steps:

- Step 1** To identify the IP addresses from which the security appliance accepts HTTPS connections, enter the following command for each address or subnet:

```
hostname(config)# http source_IP_address mask source_interface
```

- Step 2** To enable the HTTPS server, enter the following command:

```
hostname(config)# http server enable
```

- Step 3** To specify the location of the ASDM image, enter the following command:

```
hostname(config)# asdm image disk0:/asdmfile
```

For example, to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access ASDM, enter the following commands:

```
hostname(config)# crypto key generate rsa modulus 1024
hostname(config)# write mem
hostname(config)# http server enable
hostname(config)# http 192.168.1.2 255.255.255.255 inside
```

To allow all users on the 192.168.3.0 network to access ASDM on the inside interface, enter the following command:

```
hostname(config)# http 192.168.3.0 255.255.255.0 inside
```

## Configuring ASDM and WebVPN on the Same Interface

The security appliance can support both WebVPN connections and HTTPS connections for ASDM administrative sessions simultaneously on the same interface. Both HTTPS and WebVPN use port 443 by default. Therefore, to enable both HTTPS and WebVPN on the same interface, you must specify a different port number for either HTTPS or WebVPN. An alternative is to configure WebVPN and HTTPS on different interfaces.

To specify a port for HTTPS, use the *port* argument of the **http server enable** command. The following example specifies that HTTPS ASDM sessions use port 444 on the outside interface. WebVPN is also enabled on the outside interface and uses the default port (443). With this configuration, remote users initiate ASDM sessions by entering `https://<outside_ip>:444` in the browser.

```
hostname(config)# http server enable 444
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

To specify a port for WebVPN, use the **port** command from webvpn configuration mode. The next example enables WebVPN on port 444 of the outside interface. HTTPS for ASDM is also configured on the outside interface and uses the default port (443). With this configuration, remote users initiating WebVPN sessions enter `https://<outside_ip>:444` in the browser.

```
hostname(config)# http server enable
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
```

```
hostname(config-webvpn)# port 444
hostname(config-webvpn)# enable outside
```

## Configuring AAA for System Administrators

This section describes how to enable authentication and command authorization for system administrators. Before you configure AAA for system administrators, first configure the local database or AAA server according to [Chapter 13, “AAA Server and Local Database Support.”](#)

This section includes the following topics:

- [Configuring Authentication for CLI Access, page 40-5](#)
- [Configuring Authentication To Access Privileged EXEC Mode, page 40-6](#)
- [Configuring Command Authorization, page 40-7](#)
- [Configuring Command Accounting, page 40-14](#)
- [Viewing the Current Logged-In User, page 40-14](#)
- [Recovering from a Lockout, page 40-15](#)

## Configuring Authentication for CLI Access

If you enable CLI authentication, the security appliance prompts you for your username and password to log in. After you enter your information, you have access to user EXEC mode.

To enter privileged EXEC mode, enter the **enable** command or the **login** command (if you are using the local database only).

If you configure enable authentication (see the [“Configuring Authentication for the Enable Command” section on page 40-6](#)), the security appliance prompts you for your username and password. If you do not configure enable authentication, enter the system enable password when you enter the **enable** command (set by the **enable password** command). However, if you do not use enable authentication, after you enter the **enable** command, you are no longer logged in as a particular user. To maintain your username, use enable authentication.

For authentication using the local database, you can use the **login** command, which maintains the username but requires no configuration to turn on authentication.



### Note

---

Before the security appliance can authenticate a Telnet, SSH, or HTTP user, you must first configure access to the security appliance using the **telnet**, **ssh**, and **http** commands. These commands identify the IP addresses that are allowed to communicate with the security appliance.

---

To authenticate users who access the CLI, enter the following command:

```
hostname(config)# aaa authentication {telnet | ssh | http | serial} console {LOCAL |
server_group [LOCAL]}
```

The **http** keyword authenticates the ASDM client that accesses the security appliance using HTTPS. You only need to configure HTTP authentication if you want to use a AAA server. By default, ASDM uses the local database for authentication even if you do not configure this command. HTTP management authentication does not support the SDI protocol for a AAA server group.

If you use a AAA server group for authentication, you can configure the security appliance to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the security appliance prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

## Configuring Authentication To Access Privileged EXEC Mode

You can configure the security appliance to authenticate users with a AAA server or the local database when they enter the **enable** command. Alternatively, users are automatically authenticated with the local database when they enter the **login** command, which also accesses privileged EXEC mode depending on the user level in the local database.

This section includes the following topics:

- [Configuring Authentication for the Enable Command, page 40-6](#)
- [Authenticating Users Using the Login Command, page 40-6](#)

### Configuring Authentication for the Enable Command

You can configure the security appliance to authenticate users when they enter the **enable** command. If you do not authenticate the **enable** command, when you enter **enable**, the security appliance prompts for the system enable password (set by the **enable password** command), and you are no longer logged in as a particular user. Applying authentication to the **enable** command maintains the username. This feature is particularly useful when you perform command authorization, where usernames are important to determine the commands a user can enter.

To authenticate users who enter the **enable** command, enter the following command:

```
hostname(config)# aaa authentication enable console {LOCAL | server_group [LOCAL]}
```

The user is prompted for the username and password.

If you use a AAA server group for authentication, you can configure the security appliance to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the security appliance prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

### Authenticating Users Using the Login Command

From user EXEC mode, you can log in as any username in the local database using the **login** command.

This feature allows users to log in with their own username and password to access privileged EXEC mode, so you do not have to give out the system enable password to everyone. To allow users to access privileged EXEC mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower. See the “[Configuring Local Command Authorization](#)” section on page 40-8 for more information.

**Caution**

If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged EXEC mode, you should configure command authorization. Without command authorization, users can access privileged EXEC mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use a AAA server for authentication, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged EXEC mode.

To log in as a user from the local database, enter the following command:

```
hostname> login
```

The security appliance prompts for your username and password. After you enter your password, the security appliance places you in the privilege level that the local database specifies.

## Configuring Command Authorization

By default when you log in, you can access user EXEC mode, which offers only minimal commands. When you enter the **enable** command (or the **login** command when you use the local database), you can access privileged EXEC mode and advanced commands, including configuration commands. If you want to control the access to commands, the security appliance lets you configure command authorization, where you can determine which commands that are available to a user.

This section includes the following topics:

- [Command Authorization Overview, page 40-7](#)
- [Configuring Local Command Authorization, page 40-8](#)
- [Configuring TACACS+ Command Authorization, page 40-11](#)

## Command Authorization Overview

You can use one of two command authorization methods:

- Local database—Configure the command privilege levels on the security appliance. When a local user authenticates with the **enable** command (or logs in with the **login** command), the security appliance places that user in the privilege level that is defined by the local database. The user can then access commands at the user's privilege level and below.

**Note**

You can use local command authorization without any users in the local database and without CLI or enable authentication. Instead, when you enter the **enable** command, you enter the system enable password, and the security appliance places you in level 15. You can then create enable passwords for every level, so that when you enter **enable n** (2 to 15), the security appliance places you in level *n*. These levels are not used unless you turn on local command authorization (see “[Configuring Local Command Authorization](#)” below). (See the *Cisco Security Appliance Command Reference* for more information about **enable**.)

- TACACS+ server—On the TACACS+ server, configure the commands that a user or group can use after they authenticate for CLI access. Every command that a user enters at the CLI is checked with the TACACS+ server.

## Configuring Local Command Authorization

Local command authorization places each user at a privilege level, and each user can enter any command at their privilege level or below. The security appliance lets you assign commands to one of 16 privilege levels (0 to 15). By default, each command is assigned either to privilege level 0 or 15.

This section includes the following topics:

- [Local Command Authorization Prerequisites, page 40-8](#)
- [Default Command Privilege Levels, page 40-8](#)
- [Assigning Privilege Levels to Commands and Enabling Authorization, page 40-9](#)
- [Viewing Command Privilege Levels, page 40-10](#)

### Local Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure **enable** authentication. (See the “[Configuring Authentication To Access Privileged EXEC Mode](#)” section on page 40-6.)

Alternatively, you can use the **login** command (which is the same as the **enable** command with authentication), which requires no configuration. We do not recommend this option because it is not as secure as enable authentication.

You can also use CLI authentication, but it is not required.

- Configure each user in the local database at a privilege level from 0 to 15.

### Default Command Privilege Levels

By default, the following commands are assigned to privilege level 0. All other commands are at level 15.

- **show checksum**
- **show curpriv**
- **enable** (enable mode)
- **help**
- **show history**
- **login**
- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

If you move any configure mode commands to a lower level than 15, be sure to move the **configure** command to that level as well, otherwise, the user will not be able to enter configuration mode.

To view all privilege levels, see the “[Viewing Command Privilege Levels](#)” section on page 40-10.

## Assigning Privilege Levels to Commands and Enabling Authorization

To assign a command to a new privilege level, and enable authorization, follow these steps:

**Step 1** To assign a command to a privilege level, enter the following command:

```
hostname(config)# privilege [show | clear | cmd] level level [mode {enable | cmd}] command
command
```

Repeat this command for each command you want to reassign.

See the following information about the options in this command:

- **show** | **clear** | **cmd**—These optional keywords let you set the privilege only for the show, clear, or configure form of the command. The configure form of the command is typically the form that causes a configuration change, either as the unmodified command (without the **show** or **clear** prefix) or as the **no** form. If you do not use one of these keywords, all forms of the command are affected.
- **level** *level*—A level between 0 and 15.
- **mode** {**enable** | **configure**}—If a command can be entered in user EXEC/privileged EXEC mode as well as configuration mode, and the command performs different actions in each mode, you can set the privilege level for these modes separately:
  - **enable**—Specifies both user EXEC mode and privileged EXEC mode.
  - **configure**—Specifies configuration mode, accessed using the **configure terminal** command.
- **command** *command*—The command you are configuring. You can only configure the privilege level of the *main* command. For example, you can configure the level of all **aaa** commands, but not the level of the **aaa authentication** command and the **aaa authorization** command separately.

Also, you cannot configure the privilege level of subcommands separately from the main command. For example, you can configure the **context** command, but not the **allocate-interface** command, which inherits the settings from the **context** command.

**Step 2** To enable local command authorization, enter the following command:

```
hostname(config)# aaa authorization command LOCAL
```

Even if you set command privilege levels, command authorization does not take place unless you enable command authorization with this command.

For example, the **filter** command has the following forms:

- **filter** (represented by the **configure** option)
- **show running-config filter**
- **clear configure filter**

You can set the privilege level separately for each form, or set the same privilege level for all forms by omitting this option. For example, set each form separately as follows.

```
hostname(config)# privilege show level 5 command filter
hostname(config)# privilege clear level 10 command filter
hostname(config)# privilege cmd level 10 command filter
```

Alternatively, you can set all filter commands to the same level:

```
hostname(config)# privilege level 5 command filter
```

The **show privilege** command separates the forms in the display.

The following example shows the use of the **mode** keyword. The **enable** command must be entered from user EXEC mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level.

```
hostname(config)# privilege cmd level 0 mode enable command enable
hostname(config)# privilege cmd level 15 mode cmd command enable
hostname(config)# privilege show level 15 mode cmd command enable
```

This example shows an additional command, the **configure** command, that uses the **mode** keyword:

```
hostname(config)# privilege show level 5 mode cmd command configure
hostname(config)# privilege clear level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode enable command configure
```


**Note**

This last line is for the **configure terminal** command.

## Viewing Command Privilege Levels

The following commands let you view privilege levels for commands.

- To show all commands, enter the following command:

```
hostname(config)# show running-config all privilege all
```

- To show commands for a specific level, enter the following command:

```
hostname(config)# show running-config privilege level level
```

The *level* is an integer between 0 and 15.

- To show the level of a specific command, enter the following command:

```
hostname(config)# show running-config privilege command command
```

For example, for the **show running-config all privilege all** command, the system displays the current assignment of each CLI command to a privilege level. The following is sample output from the command.

```
hostname(config)# show running-config all privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
....
```

The following command displays the command assignments for privilege level 10:

```
hostname(config)# show running-config privilege level 10
privilege show level 10 command aaa
```

The following command displays the command assignment for the **access-list** command:

```
hostname(config)# show running-config privilege command access-list
```

```
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
```

## Configuring TACACS+ Command Authorization

If you enable TACACS+ command authorization, and a user enters a command at the CLI, the security appliance sends the command and username to the TACACS+ server to determine if the command is authorized.

When configuring command authorization with a TACACS+ server, do not save your configuration until you are sure it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the security appliance. If you still get locked out, see the [“Recovering from a Lockout” section on page 40-15](#).

Be sure that your TACACS+ system is completely stable and reliable. The necessary level of reliability typically requires that you have a fully redundant TACACS+ server system and fully redundant connectivity to the security appliance. For example, in your TACACS+ server pool, include one server connected to interface 1, and another to interface 2. You can also configure local command authorization as a fallback method if the TACACS+ server is unavailable. In this case, you need to configure local users and command privilege levels according to the [“Configuring Command Authorization” section on page 40-7](#).

This section includes the following topics:

- [TACACS+ Command Authorization Prerequisites, page 40-11](#)
- [Configuring Commands on the TACACS+ Server, page 40-11](#)
- [Enabling TACACS+ Command Authorization, page 40-14](#)

### TACACS+ Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure CLI authentication (see the [“Configuring Local Command Authorization” section on page 40-8](#)).
- Configure **enable** authentication (see the [“Configuring Authentication To Access Privileged EXEC Mode” section on page 40-6](#)).

### Configuring Commands on the TACACS+ Server

You can configure commands on a Cisco Secure Access Control Server (ACS) TACACS+ server as a shared profile component, for a group, or for individual users. For third-party TACACS+ servers, see your server documentation for more information about command authorization support.

See the following guidelines for configuring commands in Cisco Secure ACS Version 3.1; many of these guidelines also apply to third-party servers:

- The security appliance sends the commands to be authorized as “shell” commands, so configure the commands on the TACACS+ server as shell commands.



**Note** Cisco Secure ACS might include a command type called “pix-shell.” Do not use this type for security appliance command authorization.

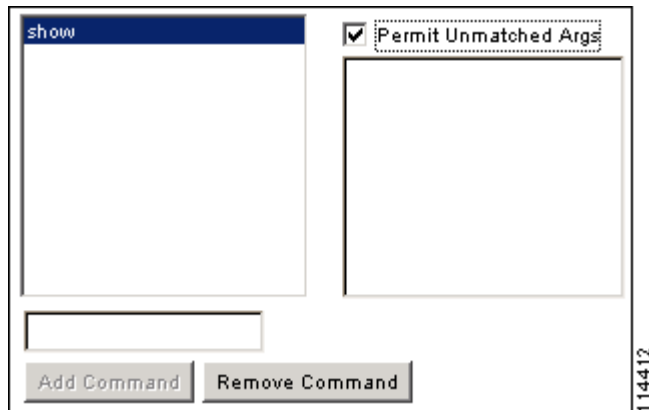
- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

For example, to allow the **show running-configuration aaa-server** command, add **show running-configuration** to the command box, and type **permit aaa-server** in the arguments box.

- You can permit all arguments of a command that you do not explicitly deny by selecting the **Permit Unmatched Args** check box.

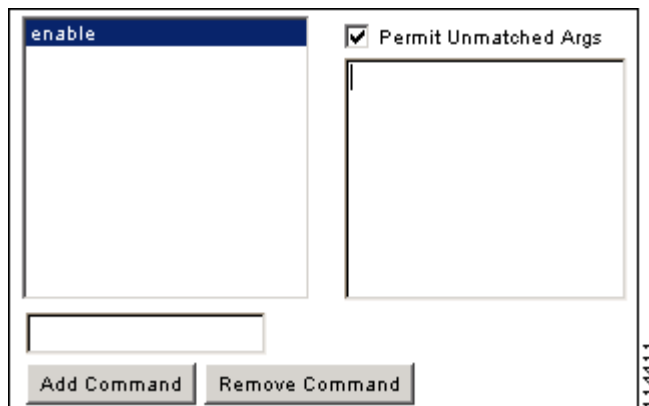
For example, you can configure just the **show** command, and then all the **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and **?**, which shows CLI usage (see [Figure 40-1](#)).

**Figure 40-1** Permitting All Related Commands



- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see [Figure 40-2](#)).

**Figure 40-2** Permitting Single Word Commands



- To disallow some arguments, enter the arguments preceded by **deny**.

For example, to allow **enable**, but not **enable password**, enter **enable** in the commands box, and **deny password** in the arguments box. Be sure to select the Permit Unmatched Args check box so that **enable** alone is still allowed (see [Figure 40-3](#)).

Figure 40-3 Disallowing Arguments

- When you abbreviate a command at the command line, the security appliance expands the prefix and main command to the full text, but it sends additional arguments to the TACACS+ server as you enter them.

For example, if you enter **sh log**, then the security appliance sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the security appliance sends **show logging mess** to the TACACS+ server, and not the expanded command **show logging message**. You can configure multiple spellings of the same argument to anticipate abbreviations (see Figure 40-4).

Figure 40-4 Specifying Abbreviations

- We recommend that you allow the following basic commands for all users:
  - **show checksum**
  - **show curpriv**
  - **enable**
  - **help**
  - **show history**
  - **login**
  - **logout**
  - **pager**

- **show pager**
- **clear pager**
- **quit**
- **show version**

## Enabling TACACS+ Command Authorization

Before you enable TACACS+ command authorization, be sure that you are logged into the security appliance as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the security appliance. For example, you should log in as an admin user with all commands authorized. Otherwise, you could become unintentionally locked out.

To perform command authorization using a TACACS+ server, enter the following command:

```
hostname(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the security appliance to use the local database as a fallback method if the TACACS+ server is unavailable. To enable fallback, specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the TACACS+ server because the security appliance prompt does not give any indication which method is being used. Be sure to configure users in the local database (see the [“Configuring Command Authorization”](#) section on page 40-7) and command privilege levels (see the [“Configuring Local Command Authorization”](#) section on page 40-8).

## Configuring Command Accounting

You can send accounting messages to the TACACS+ accounting server when you enter any command other than **show** commands at the CLI. If you customize the command privilege level using the **privilege** command (see the [“Assigning Privilege Levels to Commands and Enabling Authorization”](#) section on page 40-9), you can limit which commands the security appliance accounts for by specifying a minimum privilege level. The security appliance does not account for commands that are below the minimum privilege level.

To enable command accounting, enter the following command:

```
hostname(config)# aaa accounting command [privilege level] server-tag
```

Where *level* is the minimum privilege level and *server-tag* is the name of the TACACS+ server group that to which the security appliance should send command accounting messages. The TACACS+ server group configuration must already exist. For information about configuring a AAA server group, see the [“Identifying AAA Server Groups and Servers”](#) section on page 13-12.

## Viewing the Current Logged-In User

To view the current logged-in user, enter the following command:

```
hostname# show curpriv
```

See the following sample **show curpriv** command output. A description of each field follows.

```
hostname# show curpriv
Username : admin
Current privilege level : 15
Current Mode/s : P_PRIV
```

Table 40-1 describes the `show curpriv` command output.

**Table 40-1** *show curpriv Display Description*

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (user EXEC) or enable_15 (privileged EXEC).
Current privilege level	Level from 0 to 15. Unless you configure local command authorization and assign commands to intermediate privilege levels, levels 0 and 15 are the only levels that are used.
Current Mode/s	Shows the access modes: <ul style="list-style-type: none"> <li>• P_UNPR—User EXEC mode (levels 0 and 1)</li> <li>• P_PRIV—Privileged EXEC mode (levels 2 to 15)</li> <li>• P_CONF—Configuration mode</li> </ul>

## Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the security appliance CLI. You can usually recover access by restarting the security appliance. However, if you already saved your configuration, you might be locked out. Table 40-2 lists the common lockout conditions and how you might recover from them.

**Table 40-2** *CLI Authentication and Command Authorization Lockout Scenarios*

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users in the local database	If you have no users in the local database, you cannot log in, and you cannot add any users.	Log in and reset the passwords and <code>aaa</code> commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and add a user.
TACACS+ command authorization TACACS+ CLI authentication RADIUS CLI authentication	Server down or unreachable and you do not have the fallback method configured	If the server is unreachable, then you cannot log in or enter any commands.	<ol style="list-style-type: none"> <li>1. Log in and reset the passwords and AAA commands.</li> <li>2. Configure the local database as a fallback method so you do not get locked out when the server is down.</li> </ol>	<ol style="list-style-type: none"> <li>1. If the server is unreachable because the network configuration is incorrect on the security appliance, session into the security appliance from the switch. From the system execution space, you can change to the context and reconfigure your network settings.</li> <li>2. Configure the local database as a fallback method so you do not get locked out when the server is down.</li> </ol>

Table 40-2 CLI Authentication and Command Authorization Lockout Scenarios (continued)

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
TACACS+ command authorization	You are logged in as a user without enough privileges or as a user that does not exist	You enable command authorization, but then find that the user cannot enter any more commands.	Fix the TACACS+ server user account.  If you do not have access to the TACACS+ server and you need to configure the security appliance immediately, then log into the maintenance partition and reset the passwords and <b>aaa</b> commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and complete the configuration changes. You can also disable command authorization until you fix the TACACS+ configuration.
Local command authorization	You are logged in as a user without enough privileges	You enable command authorization, but then find that the user cannot enter any more commands.	Log in and reset the passwords and <b>aaa</b> commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and change the user level.

## Configuring a Login Banner

You can configure a message to display when a user connects to the security appliance, before a user logs in, or before a user enters privileged EXEC mode.

To configure a login banner, enter the following command in the system execution space or within a context:

```
hostname(config)# banner {exec | login | motd} text
```

Adds a banner to display at one of three times: when a user first connects (message-of-the-day (**motd**)), when a user logs in (**login**), and when a user accesses privileged EXEC mode (**exec**). When a user connects to the security appliance, the message-of-the-day banner appears first, followed by the login banner and prompts. After the user successfully logs in to the security appliance, the exec banner displays.

For the banner text, spaces are allowed but tabs cannot be entered using the CLI. You can dynamically add the hostname or domain name of the security appliance by including the strings **\$(hostname)** and **\$(domain)**. If you configure a banner in the system configuration, you can use that banner text within a context by using the **\$(system)** string in the context configuration.

To add more than one line, precede each line by the **banner** command.

For example, to add a message-of-the-day banner, enter:

```
hostname(config)# banner motd Welcome to $(hostname).
hostname(config)# banner motd Contact me at admin@example.com for any
hostname(config)# banner motd issues.
```