CHAPTER 12

Configuring AAA

Authentication, authorization, and accounting (AAA) tell the Firewall Services Module (FWSM) who the user is, what the user can do, and what the user did. This chapter contains the following sections:

Note See the "Rule Limits" section on page A-5 for information about the maximum number of AAA rules that are allowed for the entire system.

AAA Overview

AAA provides an extra level of protection and control for user access than using ACLs alone. For example, you can create an ACL allowing all outside users to access Telnet on a server on the DMZ network. If you want only some users to access the server, and you do not know their IP addresses, you can enable AAA to allow only authenticated and/or authorized users to make it through the FWSM. (The Telnet server has its own authentication; the FWSM prevents unauthorized users from attempting to access the server.)

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

This section includes the following topics:

AAA Performance

The FWSM uses "cut-through proxy" to significantly speed up performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the Open System Interconnection (OSI) model. The FWSM cut-through proxy challenges a user initially at the application layer and then authenticates against standard Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), or a local database. After the FWSM checks the policy, the FWSM shifts the session flow, and all traffic flows directly and quickly between the two parties while maintaining session state information.

About Authentication

Authentication lets you control access by requiring a valid username and password. You can configure the FWSM to authenticate the following items:

- All administrative connections to the FWSM including the following sessions:
  - Telnet
  - SSH
  - PDM (using HTTPS)
  - VPN management access (see the "Configuring VPN Client Access" section on page 11-7 for more information about using AAA with VPN)
- The **enable** command
- Network access through the FWSM

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for timeout values.) For example, if you configure the FWSM to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the session exists, the user does not also have to authenticate for FTP. See the "Authentication Overview" section on page 12-20 for more information about authentication sessions.

About Authorization

Authorization lets you control access *per user* after you authenticate with a valid username and password. You can configure the FWSM to authorize the following items:

- Management commands
- Network access through the FWSM

Authorization lets you control which services and commands are available to an individual user. Authentication alone provides the same access to services for all authenticated users.

If you need the control that authorization provides, you can configure a broad authentication rule, and then have a detailed authorization configuration. For example, you authenticate inside users to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

The FWSM caches the first 16 authorization requests per user, so if the user accesses the same services during the current authentication session, the FWSM does not resend the request to the authorization server.

About Accounting

Accounting lets you keep track of traffic that passes through the FWSM. If you enable authentication for that traffic, you can account for traffic per user. If you do not authenticate the traffic, you can account for traffic per IP address. Accounting information includes when sessions start and stop, the AAA client messages and username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

AAA Server and Local Database Support

The FWSM supports AAA servers and a local database that is stored on the FWSM. Each server type and local database provides different functionality (see Table 12-1).

Table 12-1 AAA Server and Local Database Support

Server/Database Type	Functionality	Description
RADIUS	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the RADIUS server for the username and password.
	User authentication for the **enable** command	When a user attempts to access the **enable** command, the FWSM consults the RADIUS server for the username and password.
	User authentication for network access	When a user attempts to access networks through the FWSM, and the traffic matches an authentication statement, the FWSM consults the RADIUS server for the username and password.
	User authorization for network access using downloaded ACLs per user (dynamic ACLs)	This user authorization occurs automatically when you configure authentication, but you must configure the RADIUS server to support it. When the user authenticates on the FWSM, the RADIUS server sends a dynamic ACL to the FWSM. The user's access to a given service is either permitted or denied by the ACL. The FWSM deletes the ACL when the authentication session expires.
	User authorization for network access using a downloaded ACL name per user	This user authorization occurs implicitly when you configure authentication, but you must configure the RADIUS server to support it. When the user authenticates on the FWSM, the RADIUS server sends a name of an ACL that is already defined on the FWSM. The user's access to a given service is either permitted or denied by the ACL. You can specify the same ACL for multiple users.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use a RADIUS server to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)
	Accounting for network access per user or IP address	You can configure the FWSM to send accounting information to the RADIUS server about any traffic that passes through the FWSM.

Table 12-1 AAA Server and Local Database Support (continued)

Server/Database Type	Functionality	Description
TACACS+	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the TACACS+ server for the username and password.
	User authentication for the **enable** command	When a user attempts to access the **enable** command, the FWSM consults the TACACS+ server for the username and password.
	User authentication for network access	When a user attempts to access networks through the FWSM, and the traffic matches an authentication statement, the FWSM consults the TACACS+ server for the username and password.
	User authorization for network access	When a user matches an authorization statement on the FWSM after authenticating, the FWSM consults the TACACS+ server for the user's access privileges.
	User authorization for management commands.	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.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use a TACACS+ server to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)
	Accounting for network access per user or IP address	You can configure the FWSM to send accounting information to the TACACS+ server about any traffic that passes through the FWSM.
Local database[1]	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the local user database for the username and password.
	User authentication for the **enable** or **login** command	When a user attempts to access the **enable** or **login** command, the FWSM consults the local user database for the username and password. You do not need to configure **login** user authentication; it is on by default.
	User authorization for management commands.	When a user authenticates with the **enable** command (or logs in with the **login** command), the FWSM places that user in the privilege level defined by the local database. You can configure each command to belong to privilege level between 0 and 15 on the FWSM.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use the local database to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)

1. The local database can act as a fallback method for each of these functions if the AAA server is unavailable.

Configuring the Local Database

This section describes how to manage users in the local database. You can use the local database for CLI access authentication, privileged mode authentication, command authorization, or for VPN client authentication for management access. You cannot use the local database for network access authentication or authorization. For multiple context mode, you can configure usernames in the system execution space to provide individual logins using the **login** command; however, you cannot configure any **aaa** commands in the system execution space.

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 mode, you should enable command authorization. (See the "Configuring Local Command Authorization" section on page 12-10.) Without command authorization, users can access privileged 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 RADIUS or TACACS+ authentication so the user will not be able to use the **login** command, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

To define a user account in the local database, enter the following command:

```
FWSM/contexta(config)# username username {nopassword | password password}
[privilege level]
```

Define the following parameters:

- *username*—A string from 4 to 15 characters long.
- *password*—A string from 3 to 16 characters long.
- *privilege level*—The privilege level that you want to assign to the new user account (from 0 to 15). The default is 2. This privilege level is used with command authorization.
- **nopassword**—Creates a user account with no password.

For example, the following command assigns a privilege level of 15 to the admin user account:

```
FWSM/contexta(config)# username admin password passw0rd privilege 15
```

The following command creates a user account with no password:

```
FWSM/contexta(config)# username john.doe nopassword
```

Identifying a AAA Server

If you want to use an external AAA server (RADIUS or TACACS+) for authentication, authorization, or accounting, you must first add one or more servers to a server group on the FWSM. You identify this server group name when you add AAA rules. Each server group consists of only one type of server, RADIUS or TACACS+. For multiple context mode, you can configure up to 4 servers in a maximum of 4 groups. In single mode, you can configure 16 servers in a maximum of 14 server groups.

The FWSM contacts the first server in the group. If that server is unavailable, the FWSM contacts the next server in the group, if configured. If all servers in the group are unavailable, the FWSM tries the local database if you configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the FWSM continues to try the AAA servers.

To add a server to a group, follow these steps:

Step 1 To identify the server group name and the protocol, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group protocol {radius | tacacs+}
```

Step 2 To identify the maximum number of requests to send to a AAA server in the group before trying the next server, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group max-failed-attempts number
```

The *number* can be between 1 and 5 times. The default is 3.

If you configured a fallback method using the local database (for management access only; see the "Configuring Authentication for CLI Access" section on page 12-8, the "Configuring Authentication to Access Privileged Mode" section on page 12-8, and the "Configuring TACACS+ Command Authorization" section on page 12-13 to configure the fallback mechanism), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default) so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **aaa-server deadtime** command below.

If you do not have a fallback method, the FWSM continues to retry the servers in the group.

Step 3 If you configured a fallback method, identify the amount of time the server group is marked as unresponsive after all communications attempts fail by entering the following command:

```
FWSM/contexta(config)# aaa-server server_group deadtime minutes
```

Step 4 To add a server to the group, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group (interface_name) host server_ip [key]
[timeout seconds]
```

The *key* is a case-sensitive, alphanumeric keyword of up to 127 characters that is the same value as the key on the server. Spaces are not permitted in the key, but other special characters are permitted. The key is used between the FWSM and server for encrypting data between them.

For example, to add one TACACS+ group with one primary and one backup server, and one RADIUS group with a single server, enter the following commands:

```
FWSM/contexta(config)# aaa-server AuthInbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthInbound max-failed-attempts 2
FWSM/contexta(config)# aaa-server AuthInbound deadtime 20
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.2 TheUauthKey2
FWSM/contexta(config)# aaa-server AuthOutbound protocol radius
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
```

Configuring Authentication for CLI Access

If you enable CLI authentication, the FWSM prompts you for your username and password to log in. After you enter your information, you have access to unprivileged mode.

To enter privileged 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 to Access Privileged Mode" section on page 12-8), the FWSM 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 FWSM can authenticate a Telnet, SSH, or HTTP user, you must first configure access to the FWSM using the **telnet**, **ssh**, and **http** commands. These commands identify the IP addresses that are allowed to communicate with the FWSM. See Chapter 11, "Allowing Remote Management." The only exception is when you session from the switch to the FWSM; this Telnet session is always allowed. However, you cannot authenticate the system session because the system configuration does not contain any **aaa** commands.

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

```
FWSM/contexta(config)# aaa authentication {telnet | ssh | http} console {LOCAL | server_group [LOCAL]}
```

The **http** keyword authenticates the PDM client that accesses the FWSM using HTTPS.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM 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 FWSM 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 Mode

You can configure the FWSM 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 mode depending on the user level in the local database. See the following sections for information about these methods:

- Configuring Authentication for the enable Command, page 12-9
- Authenticating Users Using the login Command, page 12-9

Configuring Authentication for the enable Command

You can configure the FWSM to authenticate users when they enter the **enable** command. If you do not authenticate the **enable** command, when you enter **enable**, the FWSM prompts for the system enable password (set by the **enable password** command), and you are no longer logged in as a particular user. Enable authentication 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:

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

The user is prompted for the username and password.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM 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 FWSM 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 unprivileged mode, you can log in as any username in the local database using the **login** command. Unlike enable authentication, this method is available in the system execution space in multiple context mode.

This feature allows users to log in with their own username and password to access privileged mode, so you do not have to give out the system enable password to everyone. To allow users to access privileged 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 12-10 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 mode, you should configure command authorization. Without command authorization, users can access privileged 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 RADIUS or TACACS+ 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 mode.

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

```
FWSM> login
```

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

Configuring Command Authorization

By default when you log in, you can access unprivileged 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 mode and advanced commands, including configuration commands. If you want to control the access to commands, the FWSM 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

You can use one of two command authorization methods:

- Local database—Configure the command privilege levels on the FWSM. When a local user authenticates with the **enable** command (or logs in with the **login** command), the FWSM 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 FWSM 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 FWSM 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 *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module 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 FWSM 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 to privilege level 15.

This section includes the following topics:

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 Mode" section on page 12-8.)

  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 (see the "Configuring Authentication for CLI Access" section on page 12-8), but it is not required.

- Configure each user in the local database at a privilege level from 0 to 15. (See the "Configuring the Local Database" section on page 12-6.)

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 12-13.

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:

```
FWSM/contexta(config)# privilege [show | clear | configure] level level
[mode {enable | configure}] command command
```

Repeat this command for each command you want to reassign.

See the following information about the options in this command:

- **show | clear | configure**—These optional keywords allow you to 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 unprivileged/privileged 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 unprivileged mode and privileged 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:

```
FWSM/contexta(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 filter**
- **clear 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:

```
FWSM/contexta(config)# privilege show level 5 command filter
FWSM/contexta(config)# privilege clear level 10 command filter
FWSM/contexta(config)# privilege configure level 10 command filter
```

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

```
FWSM/contexta(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 unprivileged mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level.

```
FWSM/contexta(config)# privilege configure level 0 mode enable command enable
FWSM/contexta(config)# privilege configure level 15 mode configure command enable
FWSM/contexta(config)# privilege show level 15 mode configure command enable
```

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

```
FWSM/contexta(config)# privilege show level 5 mode configure command configure
FWSM/contexta(config)# privilege clear level 15 mode configure command configure
```

```
FWSM/contexta(config)# privilege configure level 15 mode configure command configure
FWSM/contexta(config)# privilege configure level 15 mode enable command configure
```

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

Viewing Command Privilege Levels

The following commands allow you to view privilege levels for commands.

- To show all commands, enter the following command:

  ```
  FWSM/contexta(config)# show privilege all
  ```

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

  ```
  FWSM/contexta(config)# show privilege level level
  ```

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

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

  ```
  FWSM/contexta(config)# show privilege command command
  ```

For example, for the **show privilege all** command, the system displays the current assignment of each CLI command to a privilege level. The following example illustrates the first part of the display:

```
FWSM(config)# show 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:

```
FWSM/contexta(config)# show privilege level 10
privilege show level 10 command aaa
```

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

```
FWSM/contexta(config)# show 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 FWSM 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 FWSM. If you still get locked out, see the "Recovering from a Lockout" section on page 12-19.

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 FWSM. 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 Local Command Authorization" section on page 12-10.

This section includes the following topics:

TACACS+ Command Authorization Prerequisites

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

- Configure CLI authentication (see the "Configuring Authentication for CLI Access" section on page 12-8).
- Configure **enable** authentication (see the "Configuring Authentication to Access Privileged Mode" section on page 12-8).

Configuring Commands on the TACACS+ Server

You can configure commands on a CiscoSecure 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 on a CiscoSecure ACS TACACS+ server Version 3.1; many of these guidelines also apply to third-party servers:

- The FWSM sends the commands to be authorized as "shell" commands, so configure the commands on the TACACS+ server as shell commands.

Note The Cisco Secure ACS server might include a command type called "pix-shell." Do not use this type for FWSM 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 **show aaa**, **aaa authentication**, and **aaa authorization command** commands, add **aaa** to the command box, and type **permit authentication** and **permit authorization command** in the arguments box. The **show aaa** command must be listed separately (see Figure 12-1 and Figure 12-2).

Figure 12-1 Permitting Specific Commands

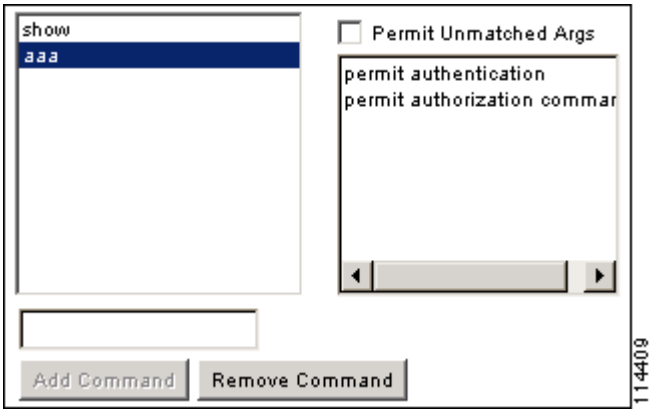
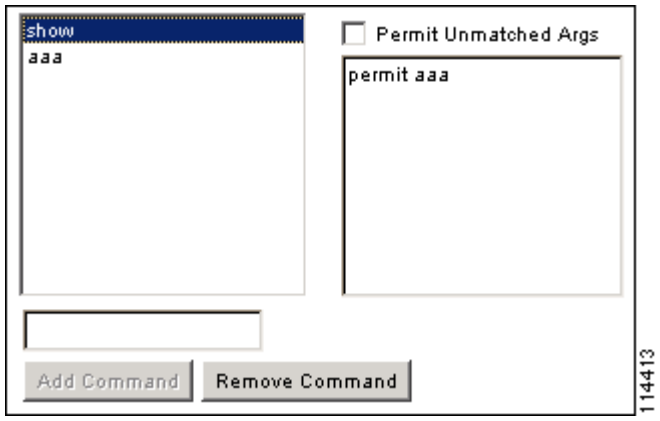


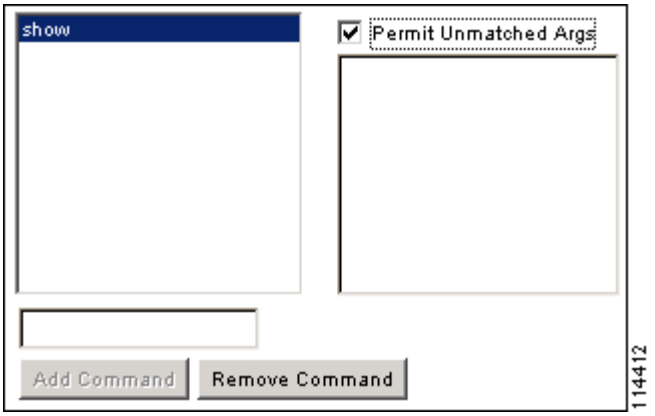
Figure 12-2 Permitting the Show Version of a Command



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

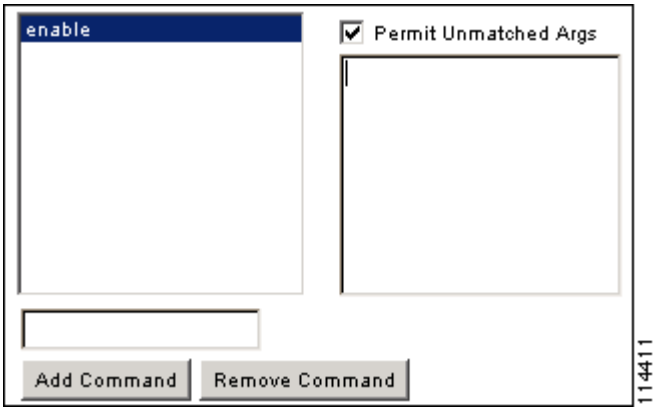
  For example, you can configure just the **show** command, and then all **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and **?**, which shows CLI usage (see Figure 12-3).

Figure 12-3 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 12-4).

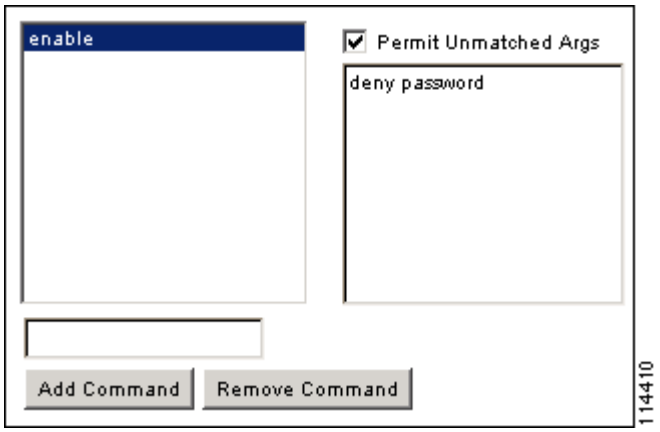
Figure 12-4 Permitting Single Word Commands



- 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 12-5).

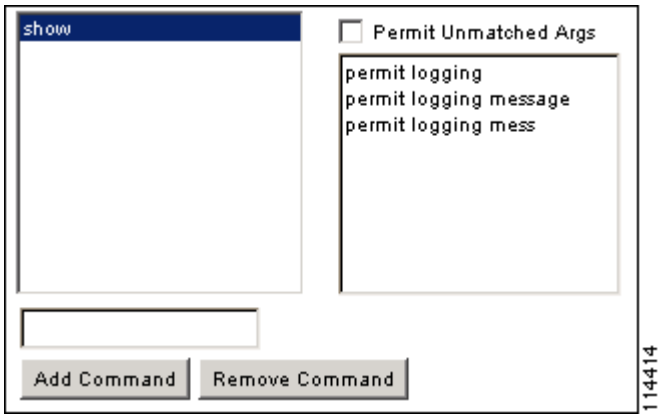
Figure 12-5 Disallowing Arguments



- When you abbreviate a command at the command line, the FWSM 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 FWSM sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the FWSM 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 12-6).

Figure 12-6 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 FWSM as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the FWSM. 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:

```
FWSM/contexta(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the FWSM 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 FWSM prompt does not give any indication which method is being used.

Be sure to configure users in the local database (see the "Configuring the Local Database" section on page 12-6) and command privilege levels (see the "Configuring Local Command Authorization" section on page 12-10).

Viewing the Current Logged-In User

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

```
FWSM/contexta# show curpriv
```

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

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

Table 12-2 describes the **show curpriv** command output.

Table 12-2 show curpriv Display Description

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (unprivileged) or enable_15 (privileged).
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: • P_UNPR—Unprivileged mode (levels 0 and 1) • P_PRIV—Privileged mode (levels 2 to 15) • P_CONF—Configuration mode

Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the FWSM CLI. You can usually recover access by restarting the FWSM. However, if you already saved your configuration, you might be locked out. Table 12-3 lists the common lockout conditions and how you might recover from them.

Table 12-3 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 into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM 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.	1. Log into the maintenance partition and reset the passwords and AAA commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.	1. If the server is unreachable because the network configuration is incorrect on the FWSM, session into the FWSM from the switch. From the system execution space, you can change to the context and reconfigure your network settings. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.

Table 12-3 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 FWSM immediately, then log into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM 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 into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM from the switch. From the system execution space, you can change to the context and change the user level.

Configuring Authentication for Network Access

This section includes the following topics:

Authentication Overview

The FWSM lets you configure network access authentication using RADIUS or TACACS+ servers.

Although you can configure network access authentication for any protocol or service, you can authenticate directly with HTTP, Telnet, or FTP only. A user must first authenticate with one of these services before other traffic that requires authentication is allowed through. If you do not want to allow HTTP, Telnet, or FTP through the FWSM, but want to authenticate other types of traffic, you can configure virtual Telnet; the user Telnets to a given IP address configured on the FWSM, and the FWSM provides a Telnet prompt. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **virtual telnet** command.

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for timeout values.) For

example, if you configure the FWSM to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

For Telnet, HTTP, and FTP, the FWSM generates an authentication prompt. If the destination server also has its own authentication, the user enters another username and password.

Note If you use HTTP authentication, the RADIUS or TACACS+ username and password are sent in clear text to the destination web server, and not just to the AAA server. Therefore, you should enable HTTP authentication with caution. For example, if you authenticate inside users when they access outside web servers, anyone on the outside can learn the user's RADIUS or TACACS+ username and password. We recommend that you use URL filtering if you want to control external web access. You can also use virtual HTTP, which allows the FWSM to authenticate HTTP users directly and then forward the requests to the final destination. This feature can have a serious impact on performance, however. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **virtual http** command.

For FTP, a user has the option of entering the FWSM username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the FWSM password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text:

```
name> john_c@jchrichton
password> letmein@he110
```

This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

Enabling Network Access Authentication

To configure authentication, enter the following command:

```
FWSM/contexta(config)# aaa authentication match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP, Telnet, or FTP in the ACL because the user must authenticate with one of these services before other services are allowed through the FWSM.

Note You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

For example, the following commands authenticate all inside HTTP traffic and SMTP traffic:

```
FWSM/contexta(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
FWSM/contexta(config)# access-list MAIL_AUTH extended permit tcp any any eq www
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
```

```
FWSM/contexta(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
```

The following commands authenticate Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthInbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

Configuring Authorization for Network Access

After a user authenticates for a given connection, the FWSM checks for an authorization rule or a dynamic ACL for the traffic. The authorization server or dynamic ACL then determines whether the traffic is allowed or denied.

The FWSM supports TACACS+ authorization servers. You identify the traffic that you want to authorize in the FWSM configuration, and the TACACS+ server determines a user's authorization based on the user profile.

Alternatively, you can use dynamic ACLs that are downloaded from a RADIUS server at the time of authentication. The configuration on the FWSM consists only of the authentication configuration; you enable downloadable ACLs on the server itself.

This section includes the following topics:

Configuring TACACS+ Authorization

The FWSM lets you configure network access authorization using TACACS+. A user first authenticates using HTTP, Telnet, or FTP. If any traffic from an authenticated user matches an authorization rule, the FWSM sends the username to the TACACS+ server. The TACACS+ server responds to the FWSM with a permit or a deny for that traffic, based on the user's profile. If a user is not yet authenticated, and the traffic matches an authorization statement, then the traffic is blocked. Any traffic that you want to be authorized must also be allowed through the FWSM by an ACL assigned to the interface; you cannot permit addresses in the authorization statement that are denied by the interface ACL.

See the TACACS+ server documentation for information about configuring network access restrictions for a user.

The authorization traffic does not need to be a subset of authentication traffic. Because a user needs to authenticate before authorization occurs, typically the authentication rule includes the same source addresses as the authorization rule. But you can configure a wider authentication rule than authorization, or a wider authorization rule than authentication (for example, a wider range of destination addresses).

To configure authorization, enter the following command:

```
FWSM/contexta(config)# aaa authorization match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization.

Note You can alternatively use the **aaa authorization include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

The following commands authenticate and authorize inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization:

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
FWSM/contexta(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

You can configure a RADIUS server to download an ACL or an ACL name to the FWSM at the time of authentication. See the "Configuring Authentication for Network Access" section on page 12-20 for more information about configuring authentication. The user is authorized to do only what is permitted in the user's ACL.

Any traffic that you want to allow according to a dynamic ACL must also be allowed through the FWSM by an ACL assigned to the interface; you cannot permit addresses in the dynamic ACL that are denied by the interface ACL. Because the dynamic ACL is only in place after you authenticate, you should require authentication for all traffic so that the dynamic ACL is in place before any traffic is allowed through the FWSM.

This section includes the following topics:

Configuring the RADIUS Server to Download Per-User Access Control Lists

This section describes how to configure a CiscoSecure ACS RADIUS server or a third-party RADIUS server, and includes the following topics:

Configuring a CiscoSecure ACS RADIUS Server for Downloadable ACLs

You can configure ACLs on the CiscoSecure ACS RADIUS server as a shared profile component and then assign the ACL to a group or to an individual user.

The ACL definition consists of one or more FWSM commands that are similar to the extended **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13), except without the following prefix:

access-list *acl_name* **extended**

The following example is an ACL definition before it is downloaded to the FWSM:

```
+-------------------------------------------+
| Shared profile Components                 |
|                                           |
|   Downloadable PIX ACLs                   |
|                                           |
| Name:               acs_ten_acl           |
| Description:  10 access-list commands  |
|                                           |
|                                           |
|      ACL Definitions                      |
|                                           |
| permit tcp any host 10.0.0.254            |
| permit udp any host 10.0.0.254            |
| permit icmp any host 10.0.0.254           |
| permit tcp any host 10.0.0.253            |
| permit udp any host 10.0.0.253            |
| permit icmp any host 10.0.0.253           |
| permit tcp any host 10.0.0.252            |
| permit udp any host 10.0.0.252            |
| permit icmp any host 10.0.0.252           |
| permit ip any any                         |
+-------------------------------------------+
```

The downloaded ACL on the FWSM has the following name:

#ACSACL#-ip-*acl_name-number*

The *acl_name* argument is the name that is defined on the RADIUS server, and *number* is a unique version ID.

The downloaded ACL on the FWSM consists of the following lines:

```
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring a Third-Party RADIUS Server for Downloadable ACLs

Configure the ACL using Cisco Vendor Specific Attribute (VSA) number 1 (cisco-AV-pair).

Configure one or more access control entries (ACEs) that are similar to the extended **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13), except that you replace the following command prefix:

access-list *acl_name* **extended**

with the following text:

ip:inacl#*nnn***=**

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the FWSM. If this parameter is omitted, the sequence value is 0, and the order in the RADIUS configuration is used.

The following example is an ACL definition before it is downloaded to the FWSM:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

The downloaded ACL name has the following format:

```
AAA-user-username
```

The *username* argument is the name of the user that is being authenticated.

The downloaded ACL on the FWSM consists of the following lines. Notice the order based on the numbers identified on the RADIUS server

```
access-list  AAA-user-john permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john deny tcp any any
access-list  AAA-user-john deny udp any any
```

Downloaded ACLs have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded ACL from a local ACL.

Configuring the RADIUS Server to Download Per-User Access Control List Names

To download a name for an ACL that you already created on the FWSM from the RADIUS server when a user authenticates, configure RADIUS attribute 11 (filter-id) as follows:

filter-id=*acl_name*

See the “Adding an Extended Access Control List” section on page 10-13 to create an ACL on the FWSM.

Configuring Accounting for Network Access

The FWSM can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the FWSM. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes when sessions start and stop, the AAA client messages and username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

To configure accounting, enter the following command:

```
FWSM/contexta(config)# aaa accounting match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for accounting, while **deny** entries exclude matching traffic from accounting.

Note You can alternatively use the **aaa accounting include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

The following commands authenticate, authorize, and account for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting:

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
FWSM/contexta(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```