



CHAPTER 22

Configuring Management Access

This chapter describes how to access the FWSM for system management through Telnet, SSH, HTTPS, and VPN. It also describes how to authenticate and authorize users.

This chapter includes the following sections:

- [Allowing Telnet Access, page 22-1](#)
- [Allowing SSH Access, page 22-2](#)
- [Allowing HTTPS Access for ASDM, page 22-4](#)
- [Allowing a VPN Management Connection, page 22-4](#)
- [Allowing ICMP to and from the FWSM, page 22-9](#)
- [AAA for System Administrators, page 22-10](#)



Note

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

The FWSM allows a maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided between all contexts. You can control the number of Telnet sessions allowed per context using resource classes. (See the [“Configuring a Class” section on page 4-24.](#)) In admin context only, you can have up to 15 Telnet and 15 SSH sessions concurrently.



Note

Please note that if you have two or more concurrent Telnet or SSH sessions and one of the sessions is at the **More** prompt, the other sessions may hang until the **More** prompt is dismissed. To disable the **More** prompt and avoid this situation, enter the **pager lines 0** command.

Please note that concurrent access to the FWSM is not recommended. In some cases, two Telnet sessions issuing the same commands might cause one of the sessions to hang until a key is depressed on the other session.

To configure Telnet access to the FWSM, perform the following steps:

- Step 1** To identify the IP addresses from which the FWSM 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 FWSM 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 FWSM, 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 FWSM on the inside interface, enter the following command:

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

Allowing SSH Access

The FWSM allows SSH connections to the FWSM for management purposes. The FWSM allows a maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided between all contexts. You can control the number of SSH sessions allowed per context using resource classes. (See the “[Configuring a Class](#)” section on page 4-24.) In admin context only, you can have up to 15 Telnet and 15 SSH sessions concurrently.



Note

Please note that if you have two or more concurrent Telnet or SSH sessions and one of the sessions is at the **More** prompt, the other sessions may hang until the **More** prompt is dismissed. To disable the **More** prompt and avoid this situation, enter the **pager lines 0** command.

SSH is an application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities. The FWSM 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 22-3](#)
- [Using an SSH Client, page 22-3](#)

Configuring SSH Access

To configure SSH access to the FWSM, perform the following 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 memory
```

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

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

The FWSM 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 FWSM 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.

- Step 5** (Optional) To restrict the version of SSH accepted by the FWSM, enter the following command. By default, the FWSM accepts both versions.

```
hostname(config)# ssh version {1 | 2}
```

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

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

Using an SSH Client

To gain access to the FWSM 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 7-1). By default, the password is “cisco.”

When starting an SSH session, a dot (.) displays on the FWSM 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 FWSM 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 FWSM. These tasks are completed if you use the **setup** command. This section describes how to manually configure ASDM access.

The FWSM allows a maximum of 5 concurrent ASDM instances per context, if available, with a maximum of 80 ASDM instances between all contexts. You can control the number of ASDM sessions allowed per context using resource classes. (See the “[Configuring a Class](#)” section on page 4-24.)

To configure ASDM access, perform the following steps:

-
- Step 1** To identify the IP addresses from which the FWSM 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
```

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)# 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
```

Allowing a VPN Management Connection

The FWSM supports IPSec for management access. An IPSec VPN ensures that IP packets can safely travel over insecure networks such as the Internet. All communication between two VPN peers occurs over a secure tunnel, which means the packets are encrypted and authenticated by the peers.

The FWSM can connect to another VPN concentrator, such as a Cisco PIX firewall or a Cisco IOS router, using a site-to-site tunnel. You specify the peer networks that can communicate over the tunnel. In the case of the FWSM, the only address available on the FWSM end of the tunnel is the interface itself.

In routed mode, the FWSM can also accept connections from VPN clients, either hosts running the Cisco VPN client, or VPN concentrators such as the Cisco PIX firewall or Cisco IOS router running the Easy VPN client. Unlike a site-to-site tunnel, you do not know in advance the IP address of the client. Instead, you rely on client authentication. Transparent firewall mode does not support remote clients. Transparent mode does support site-to-site tunnels.

The FWSM can support 5 concurrent IPSec connections, with a maximum of 10 concurrent connections divided between all contexts. You can control the number of IPSec sessions allowed per context using resource classes. (See the “Configuring a Class” section on page 4-24.)

This section describes the following topics:

- [Configuring Basic Settings for All Tunnels, page 22-5](#)
- [Configuring VPN Client Access, page 22-6](#)
- [Configuring a Site-to-Site Tunnel, page 22-8](#)

Configuring Basic Settings for All Tunnels

The following steps are required for both VPN client access and for site-to-site tunnels, and include setting the IKE policy (IKE is part of the ISAKMP) and the IPSec transforms.

To configure basic settings for all tunnels, perform the following steps:

Step 1 To set the IKE encryption algorithm, enter the following command:

```
hostname(config)# isakmp policy priority encryption {des | 3des}
```

The **3des** keyword is more secure than **des**.

You can have multiple IKE policies. The FWSM tries each policy in order of the *priority* until the policy matches the peer policy. The *priority* can be an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest. Use this same priority number for the following **isakmp** commands.

Step 2 To set the Diffie-Hellman group used for key exchange, enter the following command:

```
hostname(config)# isakmp policy priority group {1 | 2}
```

Group 1 is 768 bits, and Group 2 is 1024 bits (and therefore more secure).

Step 3 To set the authentication algorithm, enter the following command:

```
hostname(config)# isakmp policy priority hash {md5 | sha}
```

The **sha** keyword is more secure than **md5**.

Step 4 To set the IKE authentication method as a shared key, enter the following command:

```
hostname(config)# isakmp policy priority authentication pre-share
```

You can alternatively use certificates instead of a shared key by specifying the **rsa-sig** option. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about this method.

Step 5 To enable IKE on the tunnel interface, enter the following command:

```
hostname(config)# isakmp enable interface_name
```

Step 6 To set the authentication and encryption methods used for IPSec tunnels in a transform set, enter the following command:

```
hostname(config)# crypto ipsec transform-set transform_name [esp-md5-hmac | esp-sha-hmac]
{esp-aes-256 | esp-aes-192 | esp-aes | esp-des | esp-3des}
```

Although you can specify authentication alone, or encryption alone, these methods are not secure.

You refer to this transform set when you configure the VPN client group or a site-to-site tunnel.

You can refer to up to 6 transform sets for the tunnel, and the sets are checked in order until the transforms match.

The authentication and encryption algorithms of this transform typically match the IKE policy (**isakmp policy** commands). For site-to-site tunnels, this transform must match the peer transform.

Authentication options include the following (from most secure to least secure):

- **esp-sha-hmac**
- **esp-md5-hmac**

Encryption options include the following (from most secure to least secure):

- **esp-aes-256**
- **esp-aes-192**
- **esp-aes**
- **esp-3des**
- **esp-des**

Note **esp-null** (no encryption) is for testing purposes only.

For example, to configure the IKE policy and the IPsec transform sets, enter the following commands:

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform-set vpn_client esp-3des esp-sha-hmac
hostname(config)# crypto ipsec transform-set site_to_site esp-3des ah-sha-hmac
```

Configuring VPN Client Access

In routed mode, a host with Version 3.0 or 4.0 of the Cisco VPN client can connect to the FWSM for management purposes over a public network, such as the Internet.

Transparent firewall mode does not support remote clients. Transparent mode does support site-to-site tunnels.

To allow remote clients to connect to the FWSM for management access, first configure basic VPN settings (see [“Configuring Basic Settings for All Tunnels”](#)), and then perform the following steps:

-
- Step 1** To specify the transform sets (defined in the [“Configuring Basic Settings for All Tunnels”](#) section on [page 22-5](#)) allowed for client tunnels, enter the following command:

```
hostname(config)# crypto dynamic-map dynamic_map_name priority set transform-set
transform_set1 [transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first).

This dynamic crypto map allows unknown IP addresses to connect to the FWSM.

The **dynamic-map** name is used in [Step 2](#).

The *priority* specifies the order in which multiple commands are evaluated. If you have a command that specifies one set of transforms, and another that specifies others, then the priority number determines the command that is evaluated first.

- Step 2** To assign the dynamic crypto map (from [Step 1](#)) to a static tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority ipsec-isakmp dynamic
dynamic_map_name
```

- Step 3** To specify the interface at which you want the client tunnels to terminate, enter the following command:

```
hostname(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

- Step 4** To specify the range of addresses that VPN clients use on the FWSM enter the following command:

```
hostname(config)# ip local pool pool_name first_ip_address-last_ip_address [mask mask]
```

All tunneled packets from the client use one of these addresses as the source address.

- Step 5** To specify the traffic that is destined for the FWSM, so you can tunnel only that traffic according to the **tunnel group** command in [Step 7](#), enter the following command:

```
hostname(config)# access-list acl_name [extended] permit {protocol} host
fws_interface_address pool_addresses mask
```

This access list identifies traffic from the local pool (see [Step 4](#)) destined for the FWSM interface. See the “[Adding an Extended Access List](#)” section on page 12-6 for more information about access lists.

- Step 6** To assign the VPN address pool to a tunnel group, enter the following command:

```
hostname(config)# tunnel-group name general-attributes address-pool pool_name
```

This group specifies VPN characteristics for connecting clients. When a client connects to the FWSM, they need to enter the tunnel group name and password in [Step 8](#).

- Step 7** To specify that only traffic destined for the FWSM is tunneled, enter the following commands:

```
hostname(config)# group-policy name attributes
hostname(config-group-policy)# split-tunnel-policy tunnelall
```

This command is required.

- Step 8** To set the VPN group password, enter the following command:

```
hostname(config)# group-policy group_name external server-group server_group_name password
server_password
```

- Step 9** To allow Telnet or SSH access, see the “[Allowing Telnet Access](#)” section on page 22-1 and the “[Allowing SSH Access](#)” section on page 22-2.

Specify the VPN pool addresses in the **telnet** and **ssh** commands.

For example, the following commands allow VPN clients to use Telnet on the outside interface (209.165.200.225). The user authentication is the local database, so users with the tunnel group name and password, as well as the username “admin” and the password “passw0rd” can connect to the FWSM.

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
```

```

hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# username admin password passw0rd
hostname(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
hostname(config)# crypto dynamic-map vpn_client 1 set transform-set vpn
hostname(config)# crypto map telnet_tunnel 1 ipsec-isakmp dynamic vpn_client
hostname(config)# crypto map telnet_tunnel interface outside
hostname(config)# crypto map telnet_tunnel client authentication LOCAL
hostname(config)# ip local pool Firstpool 10.1.1.1-10.1.1.2
hostname(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host 10.1.1.1
hostname(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host 10.1.1.2
hostname(config)# tunnel-group StocktonAAA general-attributes address-pool Firstpool
hostname(config)# group-policy name attributes
hostname(config-group-policy)# split-tunnel-policy tunnelall
hostname(config)# group-policy ExternalGroup external server-group LodiAAA password $secure23
hostname(config)# telnet 10.1.1.1 255.255.255.255 outside
hostname(config)# telnet 10.1.1.2 255.255.255.255 outside
hostname(config)# telnet timeout 30

```

Configuring a Site-to-Site Tunnel

To configure a site-to-site tunnel, first configure basic VPN settings (see “[Configuring Basic Settings for All Tunnels](#)”), and then perform the following steps:

- Step 1** To set the shared key used by both peers, enter the following command:

```
hostname(config)# isakmp key keystring address peer-address
```

- Step 2** To identify the traffic allowed to go over the tunnel, enter the following command:

```
hostname(config)# access-list acl_name [extended] {deny | permit} {protocol} host
fwsm_interface_address dest_address mask
```

For the destination address, specify the addresses that are allowed to access the FWSM.

See the “[Adding an Extended Access List](#)” section on page 12-6 for more information about access lists.

- Step 3** To create an IPsec tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority ipsec-isakmp
```

All tunnel attributes are identified by the same **crypto map** name.

The *priority* specifies the order in which multiple commands are evaluated. If you have a command for this **crypto map** name that specifies **ipsec-isakmp**, and another that specifies **ipsec-isakmp dynamic** (for VPN client connections), then the priority number determines the command that is evaluated first.

- Step 4** To assign the access list from [Step 2](#) to this tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority match address acl_name
```

- Step 5** To specify the remote peer on which this tunnel terminates, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority set peer ip_address
```

- Step 6** To specify the transform sets for this tunnel (defined in the “[Configuring Basic Settings for All Tunnels](#)” section on page 22-5), enter the following command:

```
hostname(config)# crypto map crypto_map_name priority set transform-set transform_set1
[transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

- Step 7** To specify the interface at which you want this tunnel to terminate, enter the following command:

```
hostname(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

This command must be entered after all other **crypto map** commands. If you change any **crypto map** settings, remove this command with the **no** prefix, and reenter it.

- Step 8** To allow Telnet or SSH access, see the “[Allowing Telnet Access](#)” section on page 22-1 and the “[Allowing SSH Access](#)” section on page 22-2.

For example, the following commands allow hosts connected to the peer router (209.165.202.129) to use Telnet on the outside interface (209.165.200.225).

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
hostname(config)# isakmp key 7mfi02lirotn address 209.165.200.223
hostname(config)# access-list TUNNEL extended permit ip host 209.165.200.225 209.165.201.0
255.255.255.224
hostname(config)# crypto map telnet_tunnel 2 ipsec-isakmp
hostname(config)# crypto map telnet_tunnel 1 match address TUNNEL
hostname(config)# crypto map telnet_tunnel 1 set peer 209.165.202.129
hostname(config)# crypto map telnet_tunnel 1 set transform-set vpn
hostname(config)# crypto map telnet_tunnel interface outside
hostname(config)# telnet 209.165.201.0 255.255.255.224 outside
hostname(config)# telnet timeout 30
```

Allowing ICMP to and from the FWSM

By default, ICMP (including ping) is not allowed to an FWSM interface (or through the FWSM. To allow ICMP *through* the FWSM, see [Chapter 14, “Permitting or Denying Network Access.”](#)). ICMP is an important tool for testing your network connectivity; however, it can also be used to attack the FWSM or your network. We recommend allowing ICMP during your initial testing, but then disallowing it during normal operation.

See the “[Rule Limits](#)” section on page A-6 for information about the maximum number of ICMP rules allowed for the entire system.

To permit or deny address(es) to reach an FWSM interface with ICMP (either from a host to the FWSM, or from the FWSM to a host, which requires the ICMP reply to be allowed back), enter the following command:

```
hostname(config)# icmp {permit | deny} {host ip_address | ip_address mask | any}
[icmp_type] interface_name
```

If you do not specify an *icmp_type*, all types are identified. You can enter the number or the name. To control ping, specify **echo-reply (0)** (FWSM to host) or **echo (8)** (host to FWSM). See the “[ICMP Types](#)” section on page E-15 for a list of ICMP types.

Like access lists, the FWSM matches a packet to each **icmp** statement in order. You should use specific statements first, and general statements later. There is an implicit deny at the end. For example, if you allow all addresses first, then deny a specific address after, then that address will be unintentionally allowed because it matched the first statement.

**Note**

If you only want to allow the FWSM to ping a host (and thus allow the echo reply back to the interface), and not allow hosts to ping the FWSM, you can enable the ICMP inspection engine instead of entering the command above. See [Chapter 21, “Applying Application Layer Protocol Inspection.”](#)

For example, to allow all hosts except the one at 10.1.1.15 to use ICMP to the inside interface, enter the following commands:

```
hostname(config)# icmp deny host 10.1.1.15 inside
hostname(config)# icmp permit any inside
```

To allow the host at 10.1.1.15 to use only ping to the inside interface, enter the following commands:

```
hostname(config)# icmp permit host 10.1.1.15 inside
```

AAA for System Administrators

This section describes how to enable CLI authentication, command authorization, and command accounting for system administrators. Before you configure AAA for system administrators, first configure the local database or AAA server according to [Chapter 11, “Configuring AAA Servers and the Local Database.”](#)

**Note**

In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet authentication in the admin context, then authentication also applies to sessions from the switch to the FWSM (which enters the system execution space). See the [“Configuring Authentication for CLI and ASDM Access”](#) section on page 22-10 for more information.

This section includes the following topics:

- [Configuring Authentication for CLI and ASDM Access, page 22-10](#)
- [Configuring Authentication to Access Privileged EXEC Mode, page 22-13](#)
- [Configuring Command Authorization, page 22-14](#)
- [Configuring Command Accounting, page 22-22](#)
- [Viewing the Current Logged-In User, page 22-22](#)
- [Recovering from a Lockout, page 22-23](#)

Configuring Authentication for CLI and ASDM Access

This section explains how to configure CLI authentication when you use Telnet or SSH, and how to configure ASDM authentication. This section includes the following topics:

- [CLI Access Overview, page 22-11](#)
- [ASDM Access Overview, page 22-11](#)

- [Authenticating Sessions from the Switch to the FWSM, page 22-11](#)
- [Enabling CLI or ASDM Authentication, page 22-12](#)

CLI Access Overview

Before the FWSM can authenticate a Telnet or SSH user, you must first configure access to the FWSM using the **telnet** or **ssh** commands (see the [“Allowing Telnet Access” section on page 22-1](#) and [“Allowing SSH Access” section on page 22-2](#)). These commands identify the IP addresses that are allowed to communicate with the FWSM. The exception is for access to the system in multiple context mode; a session from the switch to the FWSM is a Telnet session, but the **telnet** command is not required.

After you connect to the FWSM, you log in and access user EXEC mode.

- If you do not enable any authentication for Telnet, you do not enter a username; you enter the login password (set with the **password** command). For SSH, you enter “pix” as the username, and enter the login password.
- If you enable Telnet or SSH authentication according to this section, you enter the username and password as defined on the AAA server or local user database.

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

- 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.
- If you configure enable authentication (see the [“Configuring Authentication for the Enable Command” section on page 22-13](#)), the FWSM prompts you for your username and password.

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

ASDM Access Overview

By default, you can log into ASDM with a blank username and the enable password set by the **enable password** command. However, if you enter a username and password at the login screen (instead of leaving the username blank), ASDM checks the local database for a match.

Although you can configure HTTP authentication according to this section and specify the local database, that functionality is always enabled by default. You should only configure HTTP authentication if you want to use a RADIUS or TACACS+ server for authentication.

Authenticating Sessions from the Switch to the FWSM

In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet authentication in the admin context, then authentication also applies to sessions from the switch to the FWSM (which enters the system execution space). The admin context AAA server or local user database are used in this instance.

Enabling CLI or ASDM Authentication

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

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

The **telnet** keyword enables authentication for Telnet sessions, and when you configure this command in the admin context, for sessioning from the switch to the FWSM.

The **ssh** keyword enables authentication for SSH sessions.

The **http** keyword authenticates the ASDM 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.

For example, to enable authentication for sessions from the switch to the FWSM system execution space, enter the following commands starting from the switch CLI:

```
Router# session slot 1 processor 1 (for an FWSM in slot 1)
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open

User Access Verification

Password: cisco (the default login password)
Type help or '?' for a list of available commands.
hostname> enable
hostname# configure terminal
hostname(config)# changeto context admin (change from the system execution space to the admin context called "admin")
hostname/admin(config)# aaa-server RADS protocol radius (adds a server group called RADS)
hostname/admin(config-aaa-server-group)# aaa-server RADS (mgmt) host 192.168.1.4 cisco (adds a RADIUS server to the RADS server group)
hostname/admin(config-aaa-server-group)# exit
hostname/admin(config)# aaa authentication telnet console RADS (enables Telnet authentication using the RADS server group)
```

The next time you session from the switch to the FWSM, you are prompted for a username and password defined on the RADIUS server:

```
Router# session slot 1 processor 1
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open

User Access Verification

Username: myRADIUSusername
Password: myRADIUSpassword
Type help or '?' for a list of available commands.
hostname>
```

Configuring Authentication to Access Privileged EXEC 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 EXEC mode depending on the user level in the local database.

This section includes the following topics:

- [Configuring Authentication for the Enable Command, page 22-13](#)
- [Authenticating Users Using the Login Command, page 22-13](#)

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 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 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 user EXEC 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. The system execution space uses the admin context local user database when you enter the **login** command; the system configuration does not contain a local user database (you cannot enter the **username** command).

The login 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 22-15](#) 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 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 EXEC mode.

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

```
hostname> 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. You can only enter the **login** command in user EXEC mode. If you are in privileged EXEC mode, enter the **disable** command to return to user EXEC mode.

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 FWSM lets you configure command authorization, where you can determine which commands are available to a user.

This section includes the following topics:

- [Command Authorization Overview, page 22-14](#)
- [Configuring Local Command Authorization, page 22-15](#)
- [Configuring TACACS+ Command Authorization, page 22-18](#)

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 privilege level and below.

You can use local command authorization without any users in the local database and without CLI or enable authentication. To do so, when you enter the **enable** command, use the system enable password, and the FWSM places you in level 15 as the default “enable_15” username. You can 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](#)”). (See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **enable** command.)

- **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.

Security Contexts and Command Authorization

The following are important points to consider when implementing command authorization with multiple security contexts:

- AAA settings are discrete per context, not shared between contexts.

When configuring command authorization, you must configure each security context separately. This provides you the opportunity to enforce different command authorizations for different security contexts.

When switching between security contexts, administrators should be aware that the commands permitted for the username specified when they login may be different in the new context session or that command authorization may not be configured at all in the new context. Failure to understand that command authorizations may differ between security contexts could confuse an administrator. This behavior is further complicated by the next point.

- New context sessions started with the **changeto** command always use the default “enable_15” username as the administrator identity, regardless of what username was used in the previous context session. This behavior can lead to confusion if command authorization is not configured for the enable_15 user or if authorizations are different for the enable_15 user than for the user in the previous context session.

This behavior also affects command accounting, which is useful only if you can accurately associate each command that is issued with a particular administrator. Because all administrators with permission to use the **changeto** command can use the enable_15 username in other contexts, command accounting records may not readily identify who was logged in as the enable_15 username. If you use different accounting servers for each context, tracking who was using the enable_15 username requires correlating the data from several servers.

When configuring command authorization, consider the following:

- An administrator with permission to use the **changeto** command effectively has permission to use all commands permitted to the enable_15 user in each of the other contexts.
- If you intend to authorize commands differently per context, ensure that in each context the enable_15 username is denied use of commands that are also denied to administrators who are permitted use of the **changeto** command.

When switching between security contexts, administrators can exit privileged EXEC mode and enter the **enable** command again to use the username they need.



Note

The system execution space does not support AAA commands; therefore, command authorization is not available in the system execution space.

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 15.

This section includes the following topics:

- [Local Command Authorization Prerequisites, page 22-16](#)
- [Default Command Privilege Levels, page 22-16](#)
- [Assigning Privilege Levels to Commands and Enabling Authorization, page 22-16](#)
- [Viewing Command Privilege Levels, page 22-18](#)

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 22-13.)

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. (See the “[Configuring the Local Database](#)” section on page 11-7.)

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**
- **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 22-18.

Assigning Privilege Levels to Commands and Enabling Authorization

To assign a command to a new privilege level and enable authorization, perform the following 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 commands that are in a configuration mode entered by the main command 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
```

The following 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
```

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

```
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 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 22-23](#).

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 Command Authorization](#)” section on [page 22-14](#).

This section includes the following topics:

- [TACACS+ Command Authorization Prerequisites](#), page 22-19
- [Configuring Commands on the TACACS+ Server](#), page 22-19
- [Enabling TACACS+ Command Authorization](#), page 22-22

TACACS+ Command Authorization Prerequisites

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

- Configure CLI authentication (see the “[Configuring Authentication for CLI and ASDM Access](#)” section on [page 22-10](#)).
- Configure **enable** authentication (see the “[Configuring Authentication to Access Privileged EXEC Mode](#)” section on [page 22-13](#)).

Configuring Commands on the TACACS+ Server

You can configure commands on a Cisco Secure Access Control Server (ACS) 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 FWSM 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 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 the **show running-configuration aaa-server** command, add **show running-configuration** to the command field, and type **permit aaa-server** in the arguments field.
- You can permit all arguments of a command that you do not explicitly deny by checking 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 22-1](#)).

Figure 22-1 Permitting All Related Commands

The screenshot shows a configuration window for the 'show' command. The 'Commands' field contains 'show'. The 'Permit Unmatched Args' checkbox is checked. Below the fields are 'Add Command' and 'Remove Command' buttons. A vertical ID number '114412' is on the right side.

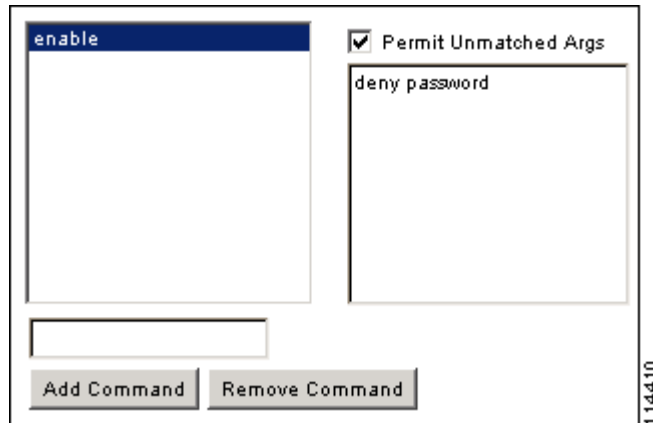
- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see Figure 22-2).

Figure 22-2 Permitting Single Word Commands

The screenshot shows a configuration window for the 'enable' command. The 'enable' command is entered in the 'Commands' field. The 'Permit Unmatched Args' checkbox is checked. Below the fields are 'Add Command' and 'Remove Command' buttons. A vertical ID number '114411' is on the right side.

- To disallow some arguments, enter the arguments preceded by **deny**.
For example, to allow **enable**, but not **enable password**, enter **enable** in the commands field, and **deny password** in the arguments field. Be sure to check the **Permit Unmatched Args** check box so that **enable** alone is still allowed (see Figure 22-3).

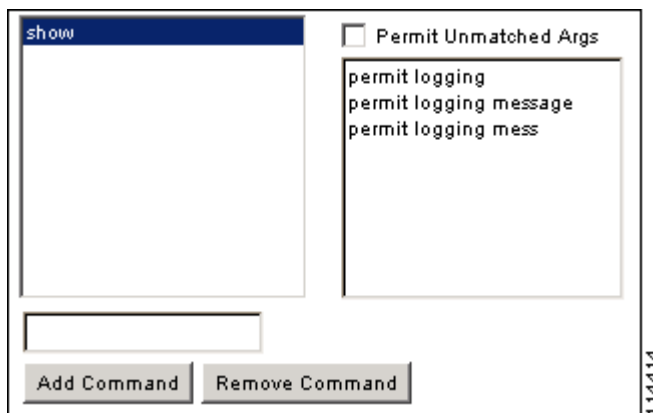
Figure 22-3 Disallowing Arguments



- 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 22-4](#)).

Figure 22-4 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**
- **quit**
- **show version**

Enabling TACACS+ Command Authorization

Before you enable TACACS+ command authorization, be sure that you are logged in to 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:

```
hostname(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 11-7) and command privilege levels (see the “[Configuring Local Command Authorization](#)” section on page 22-15).

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 22-16), you can limit which commands the FWSM accounts for by specifying a minimum privilege level. The FWSM 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 FWSM 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 11-9.

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 22-1](#) describes the **show curpriv** command output.

Table 22-1 *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"> • P_UNPR—User EXEC mode (levels 0 and 1) • P_PRIV—Privileged EXEC 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 22-2](#) lists the common lockout conditions and how you might recover from them.

Table 22-2 *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 aaa commands.	Session in to 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.	<ol style="list-style-type: none"> 1. Log in and reset the passwords and AAA commands. 2. Configure the local database as a fallback method so you do not get locked out when the server is down. 	<ol style="list-style-type: none"> 1. If the server is unreachable because the network configuration is incorrect on the FWSM, session in to 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 22-2 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 FWSM immediately, then log into the maintenance partition and reset the passwords and aaa commands.	Session in to 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 in and reset the passwords and aaa commands.	Session in to the FWSM from the switch. From the system execution space, you can change to the context and change the user level.