Configuring Management Access

This chapter describes how to access the ASA for system management through Telnet, SSH, and HTTPS (using ASDM), how to authenticate and authorize users, and how to create login banners.

This chapter includes the following sections:

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

Configuring ASA Access for ASDM, Telnet, or SSH

This section describes how to allow clients to access the ASA using ASDM, Telnet, or SSH and includes the following topics:

Licensing Requirements for ASA Access for ASDM, Telnet, or SSH

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Model Guidelines

For the ASASM, a session from the switch to the ASASM is a Telnet session, but Telnet access configuration according to this section is not required.

Additional Guidelines

- You cannot use Telnet to the lowest security interface unless you use Telnet inside a VPN tunnel.
- Management access to an interface other than the one from which you entered the ASA is not supported. For example, if your management host is located on the outside interface, you can only initiate a management connection directly to the outside interface. The only exception to this rule is through a VPN connection. See the "Configuring Management Access Over a VPN Tunnel" section on page 41-13.
- The ASA allows:
  - A maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided among all contexts.
  - A maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided among all contexts.
  - A maximum of 5 concurrent ASDM instances per context, if available, with a maximum of 32 ASDM instances among all contexts.
- The ASA supports the SSH remote shell functionality provided in SSH Versions 1 and 2 and supports DES and 3DES ciphers.
- XML management over SSL and SSH is not supported.
- (8.4 and later) The SSH default username is no longer supported. You can no longer connect to the ASA using SSH with the **pix** or **asa** username and the login password. To use SSH, you must configure AAA authentication using the **aaa authentication ssh console LOCAL** command; then define a local user by entering the **username** command. If you want to use a AAA server for authentication instead of the local database, we recommend also configuring local authentication as a backup method.

- (9.1(2) and later) The default Telnet login password was removed; you must manually set the password before using Telnet. See the "Configuring the Hostname, Domain Name, and Passwords" section on page 13-1.
- If you cannot make a Telnet or SSH connection to the ASA interface, make sure that you have enabled Telnet or SSH to the ASA according to the instructions in the "Configuring ASA Access for ASDM, Telnet, or SSH" section on page 41-1.
- The AES-CTR encryption for SSH supports only AES-128 on single-core platforms, which include the ASA 5505, 5510, 5520, 5540, and 5550.

Configuring Telnet Access

To identify the client IP addresses allowed to connect to the ASA using Telnet, perform the following steps.

Detailed Steps

	Command	Purpose
Step 1	**telnet** *source_IP_address mask source_interface* **Example:** `ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside`	For each address or subnet, identifies the IP addresses from which the ASA accepts connections. 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	**telnet timeout** *minutes* **Example:** `ciscoasa(config)# telnet timeout 30`	Sets the duration for how long a Telnet session can be idle before the ASA disconnects the session. 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 have been completed.

Examples

The following example shows how to let a host on the inside interface with an address of 192.168.1.2 access the ASA:

```
ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside
```

The following example shows how to allow all users on the 192.168.3.0 network to access the ASA on the inside interface:

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

Using a Telnet Client

To gain access to the ASA CLI using Telnet, enter the login password set by the **password** command. (9.1(2) and later) The default Telnet login password was removed; you must manually set the password before using Telnet. See the "Configuring the Hostname, Domain Name, and Passwords" section on page 13-1.

If you configure Telnet authentication (see the "Configuring Authentication for CLI and ASDM Access" section on page 41-20), then enter the username and password defined by the AAA server or local database.

Configuring SSH Access

To identify the client IP addresses and define a user allowed to connect to the ASA using SSH, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**crypto key generate rsa modulus** *modulus_size* **Example:** `ciscoasa(config)# crypto key generate rsa modulus 1024`	Generates an RSA key pair, which is required for SSH. The modulus value (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA key pair. We recommend a value of 1024.
Step 2	**write memory** **Example:** `ciscoasa(config)# write memory`	Saves the RSA keys to persistent flash memory.
Step 3	**aaa authentication ssh console LOCAL**	Enables local authentication for SSH access. You can alternatively configure authentication using a AAA server. See the "Configuring Authentication for CLI and ASDM Access" section on page 41-20 for more information.
Step 4	**username** *username* **password** *password*	Creates a user in the local database that can be used for SSH access.
Step 5	**ssh** *source_IP_address mask source_interface* **Example:** `ciscoasa(config)# ssh 192.168.3.0 255.255.255.0 inside`	For each address or subnet, identifies the IP addresses from which the ASA accepts connections, and the interface on which you can SSH. Unlike Telnet, you can SSH on the lowest security level interface.
Step 6	**ssh timeout** *minutes* **Example:** `ciscoasa(config)# ssh timeout 30`	(Optional) Sets the duration for how long an SSH session can be idle before the ASA disconnects the session. 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 have been completed.
Step 7	**ssh version** *version_number* **Example:** `ciscoasa(config)# ssh version 2`	(Optional) Limits access to SSH version 1 or 2. By default, SSH allows both versions 1 and 2.

Examples

The following example shows how to generate RSA keys and let a host on the inside interface with an address of 192.168.1.2 access the ASA:

```
ciscoasa(config)# crypto key generate rsa modulus 1024
ciscoasa(config)# write memory
ciscoasa(config)# aaa authentication ssh console LOCAL
WARNING: local database is empty! Use 'username' command to define local users.
ciscoasa(config)# username exampleuser1 password examplepassword1
ciscoasa(config)# ssh 192.168.1.2 255.255.255.255 inside
ciscoasa(config)# ssh timeout 30
```

The following example shows how to allow all users on the 192.168.3.0 network to access the ASA on the inside interface:

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

Using an SSH Client

In the SSH client on your management host, enter the username and password that you configured in the "Configuring SSH Access" section on page 41-4. When starting an SSH session, a dot (.) displays on the ASA console before the following SSH user authentication prompt appears:

```
ciscoasa(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 ASA is busy and has not hung.

You can alternatively configure a public key instead of using a password. See the "Adding a User Account to the Local Database" section on page 33-4.

Configuring HTTPS Access for ASDM

To use ASDM, you need to enable the HTTPS server, and allow HTTPS connections to the ASA. HTTPS access is enabled as part of the factory default configuration or when you use the **setup** command. This section describes how to manually configure ASDM access.

To configure HTTPS access for ASDM, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**http** *source_IP_address mask source_interface* **Example:** `ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside`	For each address or subnet, identifies the IP addresses from which the ASA accepts HTTPS connections.
Step 2	**http server enable** [*port*] **Example:** `ciscoasa(config)# http server enable 443`	Enables the HTTPS server. By default, the *port* is 443. If you change the port number, be sure to include it in the ASDM access URL. For example, if you change the port number to 444, enter the following: **https://10.1.1.1:444**

Examples

The following example shows how to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access ASDM:

```
ciscoasa(config)# http server enable
ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside
```

The following example shows how to allow all users on the 192.168.3.0 network to access ASDM on the inside interface:

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

Configuring CLI Parameters

This section includes the following topics:

Licensing Requirements for CLI Parameters

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Configuring a Login Banner

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

Restrictions

After a banner is added, Telnet or SSH sessions to ASA may close if:

- There is not enough system memory available to process the banner message(s).
- A TCP write error occurs when trying to display banner message(s).

Guidelines

- From a security perspective, it is important that your banner discourage unauthorized access. Do not use the words “welcome” or “please,” as they appear to invite intruders in. The following banner sets the correct tone for unauthorized access:

  ```
  You have logged in to a secure device. If you are not authorized to access this
  device, log out immediately or risk possible criminal consequences.
  ```

- See RFC 2196 for guidelines about banner messages.

To configure a login banner, perform the following steps:

Detailed Steps

Command	Purpose
banner {**exec** | **login** | **motd**} *text* **Example:** `ciscoasa(config)# banner motd Welcome to $(hostname).`	Adds a banner to display at one of three times: when a user first connects (message-of-the-day (**motd**)), when a user logs in (**login**), and when a user accesses privileged EXEC mode (**exec**). When a user connects to the ASA, the message-of-the-day banner appears first, followed by the login banner and prompts. After the user successfully logs in to the ASA, the exec banner appears. To add more than one line, precede each line by the **banner** command. For the banner text: • Spaces are allowed, but tabs cannot be entered using the CLI. • There are no limits for banner length other than those for RAM and flash memory. • You can dynamically add the hostname or domain name of the ASA by including the strings **$(hostname)** and **$(domain)**. • If you configure a banner in the system configuration, you can use that banner text within a context by using the **$(system)** string in the context configuration.

Examples

The following example shows how to add a message-of-the-day banner:

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

Customizing a CLI Prompt

The CLI Prompt pane lets you customize the prompt used during CLI sessions. By default, the prompt shows the hostname of the ASA. In multiple context mode, the prompt also displays the context name. You can display the following items in the CLI prompt:

cluster-unit	(Single and multiple mode) Displays the cluster unit name. Each unit in a cluster can have a unique name.
context	(Multiple mode only) Displays the name of the current context.
domain	Displays the domain name.
hostname	Displays the hostname.

priority	Displays the failover priority as pri (primary) or sec (secondary).
state	Displays the traffic-passing state of the unit. The following values appear for the state: • act—Failover is enabled, and the unit is actively passing traffic. • stby— Failover is enabled, and the unit is not passing traffic and is in a standby, failed, or another inactive state. • actNoFailover—Failover is not enabled, and the unit is actively passing traffic. • stbyNoFailover—Failover is not enabled, and the unit is not passing traffic. This condition might occur when there is an interface failure above the threshold on the standby unit. Shows the role (master or slave) of a unit in a cluster. For example, in the prompt ciscoasa/cl2/slave, the hostname is ciscoasa, the unit name is cl2, and the state name is slave.

Detailed Steps

To customize the CLI prompt, enter the following commands:

Command	Purpose
prompt {[**hostname**] [**context**] [**domain**] [**slot**] [**state**] [**priority**] [**cluster-unit**]} **Example:** `ciscoasa(config)# firewall transparent`	Customizes the CLI prompt.

Changing the Console Timeout

The console timeout sets how long a connection can remain in privileged EXEC mode or configuration mode; when the timeout is reached, the session drops into user EXEC mode. By default, the session does not time out. This setting does not affect how long you can remain connected to the console port, which never times out.

To change the console timeout, perform the following steps:

Detailed Steps

Command	Purpose
console timeout *number* **Example:** `ciscoasa(config)# console timeout 0`	Specifies the idle time in minutes (0 through 60) after which the privileged session ends. The default timeout is 0, which means the session does not time out.

Configuring ICMP Access

By default, you can send ICMP packets to any ASA interface using either IPv4 or IPv6. This section tells how to limit ICMP management access to the ASA. You can protect the ASA from attacks by limiting the addresses of hosts and networks that are allowed to have ICMP access to the ASA.

Note For allowing ICMP traffic through the ASA, see Chapter 6, "Configuring Access Rules,"in the firewall configuration guide.

This section includes the following topics:

Information About ICMP Access

ICMP in IPv6 functions the same as ICMP in IPv4. ICMPv6 generates error messages, such as ICMP destination unreachable messages and informational messages like ICMP echo request and reply messages. Additionally ICMP packets in IPv6 are used in the IPv6 neighbor discovery process and path MTU discovery.

We recommend that you always grant permission for the ICMP unreachable message type (type 3). Denying ICMP unreachable messages disables ICMP path MTU discovery, which can halt IPsec and PPTP traffic. See RFC 1195 and RFC 1435 for details about path MTU discovery.

If you configure ICMP rules, then the ASA uses a first match to the ICMP traffic followed by an implicit deny all entry. That is, if the first matched entry is a permit entry, the ICMP packet continues to be processed. If the first matched entry is a deny entry or an entry is not matched, the ASA discards the ICMP packet and generates a syslog message. An exception is when an ICMP rule is not configured; in that case, a permit statement is assumed.

Licensing Requirements for ICMP Access

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

- The ASA does not respond to ICMP echo requests directed to a broadcast address.
- The ASA only responds to ICMP traffic sent to the interface that traffic comes in on; you cannot send ICMP traffic through an interface to a far interface.
- If you cannot ping the ASA interface, make sure that you enable ICMP to the ASA for your IP address using the **icmp** command.

Default Settings

By default, you can send ICMP packets to any ASA interface using either IPv4 or IPv6.

Configuring ICMP Access

To configure ICMP access rules, enter one of the following commands:

Detailed Steps

Command	Purpose
(For IPv4) **icmp** {**permit** | **deny**} {**host** *ip_address* | *ip_address mask* | **any**} [*icmp_type*] *interface_name* **Example:** `ciscoasa(config)# icmp deny host 10.1.1.15 inside`	Creates an IPv4 ICMP access rule. 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) (ASA-to-host) or echo (8) (host-to-ASA). See the "ICMP Types" section on page 49-15 for a list of ICMP types.
(For IPv6) **ipv6 icmp** {**permit** | **deny**} {*ipv6-prefix/prefix-length* | **any** | **host** *ipv6-address*} [*icmp-type*] *interface_name* **Example:** `ciscoasa(config)# icmp permit host fe80::20d:88ff:feee:6a82 outside`	Creates an IPv6 ICMP access rule. 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) (ASA-to-host) or echo (8) (host-to-ASA). See the"ICMP Types" section on page 49-15 for a list of ICMP types.

Examples

The following example shows how to allow all hosts except the one at 10.1.1.15 to use ICMP to the inside interface:

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

The following example shows how to allow the host at 10.1.1.15 to use only ping to the inside interface, enter the following command:

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

The following example shows how to deny all ping requests and permit all packet-too-big messages (to support path MTU discovery) at the outside interface:

```
hostname(config)# ipv6 icmp deny any echo-reply outside
hostname(config)# ipv6 icmp permit any packet-too-big outside
```

The following example shows how to permit host 2000:0:0:4::2 or hosts on prefix 2001::/64 to ping the outside interface:

```
hostname(config)# ipv6 icmp permit host 2000:0:0:4::2 echo-reply outside
hostname(config)# ipv6 icmp permit 2001::/64 echo-reply outside
hostname(config)# ipv6 icmp permit any packet-too-big outside
```

Configuring Management Access Over a VPN Tunnel

If your VPN tunnel terminates on one interface, but you want to manage the ASA by accessing a different interface, you can identify that interface as a management-access interface. For example, if you enter the ASA from the outside interface, this feature lets you connect to the inside interface using ASDM, SSH, Telnet, or SNMP; or you can ping the inside interface when entering from the outside interface. Management access is available via the following VPN tunnel types: IPsec clients, IPsec site-to-site, and the AnyConnect SSL VPN client.

This section includes the following topics:

Licensing Requirements for a Management Interface

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single mode.

Firewall Mode Guidelines

Supported in routed mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

You can define only one management access interface.

Configuring a Management Interface

To configure the management interface, perform the following steps.

Detailed Steps

Command	Purpose
management-access *management_interface* **Example:** `ciscoasa(config)# management-access inside`	The *management_interface* specifies the name of the management interface that you want to access when entering the ASA from another interface.

Configuring AAA for System Administrators

This section describes how to enable authentication and command authorization for system administrators.

Information About AAA for System Administrators

This section describes AAA for system administrators and includes the following topics:

Information About Management Authentication

This section describes authentication for management access and includes the following topics:

Comparing CLI Access with and without Authentication

How you log into the ASA depends on whether or not you enable authentication:

- No Authentication—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). (SSH is not available without authentication). You access user EXEC mode.
- Authentication—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. You access user EXEC mode.

To enter privileged EXEC mode after logging in, enter the **enable** command. How **enable** works depends on whether you enable authentication:

- No Authentication—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.
- Authentication—If you configure enable authentication (see the Configuring Authentication to Access Privileged EXEC Mode (the enable Command), page 41-21), the ASA prompts you for your username and password again. This feature is particularly useful when you perform command authorization, in which usernames are important in determining the commands that a user can enter.

For enable authentication using the local database, you can use the **login** command instead of the **enable** command. **login** maintains the username but requires no configuration to turn on authentication. See the "Authenticating Users with the login Command" section on page 41-22 for more information.

Comparing ASDM Access with and without Authentication

By default, you can log into ASDM with a blank username and the enable password set by the **enable password** command. Note that 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.

If you configure HTTP authentication, you can no longer use ASDM with a blank username and the enable password.

Authenticating Sessions from the Switch to the ASA Services Module

For sessions from the switch to the ASASM (using the **session** command), you can configure Telnet authentication. For virtual console connections from the switch to the ASASM (using the **service-module session** command), you can configure serial port authentication.

In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet or serial authentication in the admin context, then authentication also applies to sessions from the switch to the ASASM. The admin context AAA server or local user database is used in this instance.

Information About Command Authorization

This section describes command authorization and includes the following topics:

Supported Command Authorization Methods

You can use one of two command authorization methods:

- Local privilege levels—Configure the command privilege levels on the ASA. When a local, RADIUS, or LDAP (if you map LDAP attributes to RADIUS attributes) user authenticates for CLI access, the ASA places that user in the privilege level that is defined by the local database, RADIUS, or LDAP server. The user can access commands at the assigned privilege level and below. Note that all users access user EXEC mode when they first log in (commands at level 0 or 1). The user needs to authenticate again with the **enable** command to access privileged EXEC mode (commands at level 2 or higher), or they can log in with the **login** command (local database only).

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 ASA 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 ASA places you in level *n*. These levels are not used unless you enable local command authorization (see the "Configuring Local Command Authorization" section on page 41-27). (See the command reference for more information about the **enable** command.)

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

About Preserving User Credentials

When a user logs into the ASA, that user is required to provide a username and password for authentication. The ASA retains these session credentials in case further authentication is needed later in the session.

When the following configurations are in place, a user needs only to authenticate with the local server for login. Subsequent serial authorization uses the saved credentials. The user is also prompted for the privilege level 15 password. When exiting privileged mode, the user is authenticated again. User credentials are not retained in privileged mode.

- The local server is configured to authenticate user access.
- Privilege level 15 command access is configured to require a password.
- The user account is configured for serial-only authorization (no access to console or ASDM).
- The user account is configured for privilege level 15 command access.

The following table shows how credentials are used in this case by the ASA.

Credentials required	Username and Password Authentication	Serial Authorization	Privileged Mode Command Authorization	Privileged Mode Exit Authorization
Username	Yes	No	No	Yes
Password	Yes	No	No	Yes
Privileged Mode Password	No	No	Yes	No

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 among contexts.

  When configuring command authorization, you must configure each security context separately. This configuration 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 which 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 that they need.

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

Licensing Requirements for AAA for System Administrators

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Prerequisites

Prerequisites for the AAA Server or Local Database

You must configure users in a AAA server or the local database. For a AAA server, you then need to configure the ASA to communicate with it. See the following chapters:

- AAA server—See the applicable AAA server-type chapter.
- Local Database—See the "Adding a User Account to the Local Database" section on page 33-4.

Prerequisites for Management Authentication

Before the ASA can authenticate a Telnet, SSH, or HTTP user, you must identify the IP addresses that are allowed to communicate with the ASA. For the ASASM, the exception is for access to the system in multiple context mode; a session from the switch to the ASASM is a Telnet session, but Telnet access configuration is not required. For more information, see the "Configuring ASA Access for ASDM, Telnet, or SSH" section on page 41-1.

Prerequisites for Local Command Authorization

- Configure **enable** authentication. (See the "Configuring Authentication for CLI and ASDM Access" section on page 41-20.)

  enable authentication is essential for maintaining the username after the user accesses the **enable** command.

  Alternatively, you can use the **login** command (which is the same as the **enable** command with authentication; for the local database only), 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.

- See the following prerequisites for each user type:
  - Local database users—Configure each user in the local database at a privilege level from 0 to 15.
  - RADIUS users—Configure the user with Cisco VSA CVPN3000-Privilege-Level with a value between 0 and 15.

– LDAP users—Configure the user with a privilege level between 0 and 15, and then map the LDAP attribute to Cisco VSA CVPN3000-Privilege-Level according to the "Configuring LDAP Attribute Maps" section on page 36-5.

Prerequisites for TACACS+ Command Authorization

- Configure CLI authentication (see the "Configuring Authentication for CLI and ASDM Access" section on page 41-20).
- Configure **enable** authentication (see the "Configuring Authentication to Access Privileged EXEC Mode (the enable Command)" section on page 41-21).

Prerequisites for Management Accounting

- Configure CLI authentication (see the "Configuring Authentication for CLI and ASDM Access" section on page 41-20).
- Configure **enable** authentication (see the "Configuring Authentication to Access Privileged EXEC Mode (the enable Command)" section on page 41-21).

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Default Settings

Default Command Privilege Levels

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

- **show checksum**
- **show curpriv**
- **enable**
- **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 Local Command Privilege Levels" section on page 41-31.

Configuring Authentication for CLI and ASDM Access

You can require authentication for CLI, ASDM, and enable command access.

Prerequisites

- Configure Telnet, SSH, or HTTP access according to the "Configuring ASA Access for ASDM, Telnet, or SSH" section on page 41-1.
- For SSH access, you must configure SSH authentication; there is no default username.

Detailed Steps

	Command	Purpose

Step 1	**aaa authentication** {**telnet** | **ssh** | **http** | **serial**} **console** {**LOCAL** | *server_group* [**LOCAL**]}
	
	Example:
	`ciscoasa(config)# aaa authentication ssh console radius_1 LOCAL`
	`ciscoasa(config)# aaa authentication http console radius_1 LOCAL`
	`ciscoasa(config)# aaa authentication serial console LOCAL`
		Authenticates users for management access. The **telnet** keyword controls Telnet access. For the ASASM, this keyword also affects the session from the switch using the **session** command. For multiple mode access, see the "Authenticating Sessions from the Switch to the ASA Services Module" section on page 41-16.
		
		The **ssh** keyword controls SSH access.
		
		The **http** keyword controls ASDM access.
		
		The **serial** keyword controls console port access. For the ASASM, this keyword affects the virtual console accessed from the switch using the **service-module session** command. For multiple mode access, see the "Authenticating Sessions from the Switch to the ASA Services Module" section on page 41-16.
		
		HTTP management authentication does not support the SDI protocol for a AAA server group.
		
		If you use a AAA server group for authentication, you can configure the ASA 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 ASA prompt does not give any indication which method is being used.
		
		You can alternatively use the local database as your primary method of authentication (with no fallback) by entering **LOCAL** alone.

Step 2	**http authentication-certificate** *interface* **Example:** `http authentication-certificate inside`	Requires a certificate from ASDM clients connecting over HTTP on the specified interface. This command can be used in addition to the **aaa authentication** command for ASDM. This command is only for ASDM access, use the command **ssl certificate-authentication** to require a certificate for all other SSL traffic, for example, cut-through proxy.

Configuring Authentication to Access Privileged EXEC Mode (the enable Command)

You can configure the ASA 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

You can configure the ASA to authenticate users when they enter the **enable** command. See the "Comparing CLI Access with and without Authentication" section on page 41-15 for more information.

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

Command	Purpose
aaa authentication enable console {**LOCAL** | *server_group* [**LOCAL**]} **Example:** `ciscoasa(config)# aaa authentication enable console LOCAL`	Authenticates users who enter the **enable** command. The user is prompted for the username and password. If you use a AAA server group for authentication, you can configure the ASA 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 ASA prompt does not give any indication of which method is being used. You can alternatively use the local database as your primary method of authentication (with no fallback) by entering **LOCAL** alone.

Authenticating Users with the login Command

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

This feature allows users to log in with their own username and password to access privileged EXEC mode, so you do not have to provide 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 41-27 for more information.

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

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

Command	Purpose
login **Example:** `ciscoasa# login`	Logs in as a user from the local database. The ASA prompts for your username and password. After you enter your password, the ASA places you in the privilege level that the local database specifies.

Limiting User CLI and ASDM Access with Management Authorization

The ASA enables you to distinguish between administrative and remote-access users when they authenticate using RADIUS, LDAP, TACACS+, or the local user database. User role differentiation can prevent remote access VPN and network access users from establishing an administrative connection to the ASA.

Note Serial access is not included in management authorization, so if you configure the **aaa authentication serial console** command, then any user who authenticates can access the console port.

Detailed Steps

Step 1 To enable management authorization for local, RADIUS, LDAP (mapped), and TACACS+ users, enter the following command:

```
ciscoasa(config)# aaa authorization exec {authentication-server | LOCAL}
```

When the **LOCAL** option is configured, the local user database is the source for the username entered and the Service-Type and Privilege-Level attributes assigned.

This option also enables support of administrative user privilege levels from RADIUS, which can be used in conjunction with local command privilege levels for command authorization. See the "Configuring Local Command Authorization" section on page 41-27 for more information.

When the **authentication-server** option is configured, the same server is used for both authentication and authorization.

Step 2 To configure the user for management authorization, see the following requirements for each AAA server type or local user:

RADIUS or LDAP (mapped) users

When users are authenticated through LDAP, the native LDAP attributes and their values can be mapped to Cisco ASA attributes to provide specific authorization features. Configure Cisco VSA CVPN3000-Privilege-Level with a value between 0 and 15. and then map the LDAP attributes to Cisco VAS CVPN3000-Privilege-Level using the **ldap map-attributes** command. For more information, see the "Configuring LDAP Attribute Maps" section on page 36-5.

The RADIUS IETF **service-type** attribute, when sent in an access-accept message as the result of a RADIUS authentication and authorization request, is used to designate which type of service is granted to the authenticated user:

- Service-Type 6 (Administrative)—Allows full access to any services specified by the **aaa authentication console** commands.
- Service-Type 7 (NAS prompt)—Allows access to the CLI when you configure the **aaa authentication {telnet | ssh} console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you configure **enable** authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command. The Framed (2) and Login (1) service types are treated the same way.
- Service-Type 5 (Outbound)—Denies management access. The user cannot use any services specified by the **aaa authentication console** commands(excluding the **serial** keyword; serial access is allowed). Remote access (IPsec and SSL) users can still authenticate and terminate their remote access sessions. All other service types (Voice, FAX, and so on) are treated the same way.

The RADIUS Cisco VSA **privilege-level** attribute (Vendor ID 3076, sub-ID 220), when sent in an access-accept message, is used to designate the level of privilege for the user.

When an authenticated user tries administrative access to the ASA through ASDM, SSH, or Telnet, but does not have the appropriate privilege level to do so, the ASA generates syslog message 113021. This message informs the user that the attempted login failed because of inappropriate administrative privileges.

The following example shows how to define an LDAP attribute map. In this example, the security policy specifies that users being authenticated through LDAP map the user record fields or parameters title and company to the IETF-RADIUS service-type and privilege-level, respectively.

```
ciscoasa(config)# ldap attribute-map admin-control
ciscoasa(config-ldap-attribute-map)# map-name title IETF-RADIUS-Service-Type
ciscoasa(config-ldap-attribute-map)# map-name company Privilege-Level
```

The following example applies an LDAP attribute map to an LDAP AAA server:

```
ciscoasa(config)# aaa-server ldap-server (dmz1) host 10.20.30.1
ciscoasa(config-aaa-server-host)# ldap-attribute-map admin-control
```

TACACS+ users

Authorization is requested with "service=shell," and the server responds with PASS or FAIL.

- PASS, privilege level 1—Allows access to ASDM, with limited read-only access to the configuration and monitoring sections, and access for **show** commands that are privilege level 1 only.
- PASS, privilege level 2 and higher—Allows access to the CLI when you configure the **aaa authentication** {**telnet** | **ssh**} **console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you configure **enable** authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command. You are not allowed to access privileged EXEC mode using the **enable** command if your enable privilege level is set to 14 or less.
- FAIL—Denies management access. You cannot use any services specified by the **aaa authentication console** commands(excluding the **serial** keyword; serial access is allowed).

Local users

Set the **service-type** command for a given username. By default, the **service-type** is **admin**, which allows full access to any services specified by the **aaa authentication console** command. For more information, see the "Adding a User Account to the Local Database" section on page 33-4.

Configuring a Password Policy for Local Database Users

When you configure authentication for CLI or ASDM access using the local database, you can configure a password policy that requires a user to change their password after a specified amount of time and also requires password standards such as a minimum length and the minimum number of changed characters.

The password policy only applies to administrative users using the local database, and not to other types of traffic that can use the local database, such as VPN or AAA for network access, and not to users authenticated by a AAA server.

This section includes the following topics:

Configuring the Password Policy

After you configure the password policy, when you change a password (either your own or another user's), the password policy applies to the new password. Any existing passwords are grandfathered in. The new policy applies to changing the password with the **username** command as well as the **change-password** command.

Prerequisites

- Configure CLI/ASDM authentication according to the "Configuring Authentication for CLI and ASDM Access" section on page 41-20. Be sure to specify the local database.
- Configure enable authentication according to the "Configuring Authentication to Access Privileged EXEC Mode (the enable Command)" section on page 41-21. Be sure to specify the local database.

Detailed Steps

	Command	Purpose
Step 1	**password-policy lifetime** *days* **Example:** `ciscoasa(config)# password-policy lifetime 180`	(Optional) Sets the interval in days after which passwords expire for remote users (SSH, Telnet, HTTP); users at the console port are never locked out due to password expiration. Valid values are between 0 and 65536 days. The default value is 0 days, a value indicating that passwords will never expire. 7 days before the password expires, a warning message appears. After the password expires, system access is denied to remote users. To gain access after expiration, do one of the following: • Have another administrator change your password with the **username** command. • Log in to the physical console port to change your password.
Step 2	**password-policy minimum-changes** *value* **Example:** `ciscoasa(config)# password-policy minimum-changes 2`	(Optional) Sets the minimum number of characters that you must change between new and old passwords. Valid values are between 0 and 64 characters. The default value is 0. Character matching is position independent, meaning that new password characters are considered changed only if they do not appear anywhere in the current password.
Step 3	**password-policy minimum-length** *value* **Example:** `ciscoasa(config)# password-policy minimum-length 8`	(Optional) Sets the minimum length of passwords. Valid values are between 3 and 64 characters. We recommend a minimum password length of 8 characters.

	Command	Purpose
Step 4	**password-policy minimum-uppercase** *value* **Example:** `ciscoasa(config)# password-policy minimum-uppercase 3`	(Optional) Sets the minimum number of upper case characters that passwords must have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 5	**password-policy minimum-lowercase** *value* **Example:** `ciscoasa(config)# password-policy minimum-lowercase 6`	(Optional) Sets the minimum number of lower case characters that passwords must have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 6	**password-policy minimum-numeric** *value* **Example:** `ciscoasa(config)# password-policy minimum-numeric 1`	(Optional) Sets the minimum number of numeric characters that passwords must have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 7	**password-policy minimum-special** *value* **Example:** `ciscoasa(config)# password-policy minimum-special 2`	(Optional) Sets the minimum number of special characters that passwords must have. Valid values are between 0 and 64 characters. Special characters include the following: !, @, #, $, %, ^, &, *, '(‘ and ‘)’. The default value is 0, which means there is no minimum.

Step 8	**password-policy authenticate enable**
	
	Example:
	`ciscoasa(config)# password-policy authenticate enable`
		(Optional) Sets whether users must change their password using the **change-password** command, instead of letting users change their password with the **username** command. The default setting is disabled: a user can use either method to change their password.
		
		If you enable this feature, if you try to change your password with the **username** command, the following error message appears:
		
		`ERROR: Changing your own password is prohibited`
		
		You also cannot delete your own account with the **clear configure username** command. If you try, the following error message appears:
		
		`ERROR: You cannot delete all usernames because you are not allowed to delete yourself`

Changing Your Password

If you configure a password lifetime in the password policy, you need to change your **username** password to a new one when the old password expires. This password change method is required if you enable password policy authentication (the **password-policy authenticate enable** command). If password policy authentication is not enabled, then you can use this method, or you can change your user account directly with the **username** command.

Detailed Steps

Command	Purpose
change-password [**old-password** *old_password* [**new-password** *new_password*]] **Example:** `hostname# change-password old-password j0hncr1chton new-password a3rynsun`	Changes your **username** password. If you do not enter the old and new passwords in the command, the ASA prompts you for input.

Configuring Command Authorization

If you want to control access to commands, the ASA lets you configure command authorization, where you can determine which commands that are available to a user. 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.

You can use one of two command authorization methods:

- Local privilege levels
- TACACS+ server privilege levels

For more information about command authorization, see the "Information About Command Authorization" section on page 41-16.

This section includes the following topics:

Configuring Local Command Authorization

Local command authorization 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. You can define each user to be at a specific privilege level, and each user can enter any command at the assigned privilege level or below. The ASA supports user privilege levels defined in the local database, a RADIUS server, or an LDAP server (if you map LDAP attributes to RADIUS attributes). See the following sections for more information:

- "Adding a User Account to the Local Database" section on page 33-4

- "Supported Authentication Methods" section on page 34-1
- "Configuring LDAP Attribute Maps" section on page 36-5

To configure local command authorization, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**privilege** [**show** | **clear** | **cmd**] **level** *level* [**mode** {**enable** | **cmd**}] **command** *command* **Example:** `ciscoasa(config)# privilege show level 5 command filter`	Assigns a command to a privilege level. Repeat this command for each command that you want to reassign. The options in this command are the following: • **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 or 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.

	Command	Purpose
Step 2	**aaa authorization exec authentication-server** **Example:** `ciscoasa(config)# aaa authorization exec authentication-server`	Supports administrative user privilege levels from RADIUS. Enforces user-specific access levels for users who authenticate for management access (see the **aaa authentication console LOCAL** command). Without this command, the ASA only supports privilege levels for local database users and defaults all other types of users to level 15. This command also enables management authorization for local, RADIUS, LDAP (mapped), and TACACS+ users. Use the **aaa authorization exec LOCAL** command to enable attributes to be taken from the local database. See the "Limiting User CLI and ASDM Access with Management Authorization" section on page 41-23 for information about configuring a user on a AAA server to accommodate management authorization.
Step 3	**aaa authorization command LOCAL** **Example:** `ciscoasa(config)# aaa authorization command LOCAL`	Enables the use of local command privilege levels, which can be checked with the privilege level of users in the local database, RADIUS server, or LDAP server (with mapped attributes). When you set command privilege levels, command authorization does not occur unless you configure command authorization with this command.

Examples

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. The following example shows how to set each form separately:

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

Alternatively, the following example shows how to set all filter commands to the same level:

```
ciscoasa(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:

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

The following example shows an additional command, the **configure** command, which uses the **mode** keyword:

```
ciscoasa(config)# privilege show level 5 mode cmd command configure
```

```
ciscoasa(config)# privilege clear level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode enable command configure
```

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

Viewing Local Command Privilege Levels

The following commandslet you view privilege levels for commands.

Command	Purpose
show running-config all privilege all	Shows all commands.
show running-config privilege level *level*	Shows commands for a specific level. The *level* is an integer between 0 and 15.
show running-config privilege command *command*	Shows the level of a specific command.

Examples

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

```
ciscoasa(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 example shows the command assignments for privilege level 10:

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

The following example shows the command assignments for the **access-list** command:

```
ciscoasa(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 Commands on the TACACS+ Server

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

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

- The ASA 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 ASA 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, 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 a question mark, which shows CLI usage.

- 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**.

- 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.

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

- 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**

Configuring TACACS+ Command Authorization

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

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

Do not save your configuration until you are sure that it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the ASA. If you still get locked out, see the "Recovering from a Lockout" section on page 41-36.

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 ASA. 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 procedures listed in the "Configuring Command Authorization" section on page 41-27.

To configure TACACS+ command authorization, enter the following command:

Detailed Steps

Command	Purpose
aaa authorization command *tacacs+_server_group* [**LOCAL**] **Example:** `ciscoasa(config)# aaa authorization command group_1 LOCAL`	Performs command authorization using a TACACS+ server. You can configure the ASA 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 ASA prompt does not give any indication which method is being used. Be sure to configure users in the local database (see the "Adding a User Account to the Local Database" section on page 33-4) and command privilege levels (see the "Configuring Local Command Authorization" section on page 41-27).

Configuring Management Access Accounting

You can send accounting messages to the TACACS+ accounting server when you enter any command other than **show** commands at the CLI. You can configure accounting when users log in, when they enter the **enable** command, or when they issue commands.

For command accounting, you can only use TACACS+ servers.

To configure management access and enable command accounting, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**aaa accounting** {**serial** | **telnet** | **ssh** | **enable**} **console** *server-tag* **Example:** `ciscoasa(config)# aaa accounting telnet console group_1`	Enables support for AAA accounting for administrative access. Valid server group protocols are RADIUS and TACACS+.
Step 2	**aaa accounting command** [**privilege** *level*] *server-tag* **Example:** `ciscoasa(config)# aaa accounting command privilege 15 group_1`	Enables command accounting. Only TACACS+ servers support command accounting. Where **privilege** *level* is the minimum privilege level and *server-tag* is the name of the TACACS+ server group to which the ASA should send command accounting messages.

Viewing the Currently Logged-In User

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

```
ciscoasa# show curpriv
```

Examples

The following is sample output from the **show curpriv** command:

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

Table 41-1 describes the **show curpriv** command output.

Table 41-1 show curpriv Command Output 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	Levels range 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 Modes	The available access modes are the following: • P_UNPR—User EXEC mode (levels 0 and 1) • P_PRIV—Privileged EXEC mode (levels 2 to 15) • P_CONF—Configuration mode

Setting a Management Session Quota

You can establish a maximum number of simultaneous management sessions. If the maximum is reached, no additional sessions are allowed and a syslog message is generated. To prevent a system lockout, the management session quota mechanism cannot block a console session.

To set a management session quota, enter the following command:

Command	Purpose
quota management-session *number* **Example:** `hostname(config)# quota management-session 1000`	Sets the maximum number of simultaneous ASDM, SSH, and Telnet sessions that are allowed on the ASA. The **no** form of this command sets the quota value to 0, which means that there is no session limit.

Exchanging Keys in an SSH Session

The Diffie-Hellman (DH) key exchange provides a shared secret that cannot be determined by either party alone. The key exchange is combined with a signature and the host key to provide host authentication. This key-exchange method provides explicit server authentication.

Both the DH Group 1 and Group 14 key-exchange methods for key exchange are supported on the ASA. If no DH group key-exchange method is specified, the DH group 1 key-exchange method is used. For more information about using DH key-exchange methods, see RFC 4253.

To exchange keys in an SSH session, enter the following command:

Command	Purpose

ssh key-exchange group {**dh-group1** | **dh-group14**} **sha-1**
 	
Example:
`ciscoasa(config)# ssh key-exchange group dh-group14 sha-1`
`ciscoasa# show running-config key-exchange`
`ssh key-exchange dh-group14-sha1`
	Exchanges keys using either the DH Group 1 or DH Group 14 key-exchange method.
	
	The **key-exchange** keyword specifies that either the DH group 1 or DH group 14 key-exchange method will follow and should be used when exchanging keys.
	
	The **group** keyword indicates that either the DH group 1 key-exchange method or the DH group 14 key-exchange method will follow and should be used when exchanging keys.
	
	The **dh-group1** keyword indicates that the DH group 1 key-exchange method will follow and should be used when exchanging keys. DH group 2 is called DH group 1 for legacy reasons.
	
	The **dh-group14** keyword indicates that the DH group 14 key-exchange method will follow and should be used when exchanging keys.
	
	The **sha-1** keyword indicates that the SHA-1 encryption algorithm should be used.
	
	Use the **show running-config ssh key-exchange** command to display the DH group key-exchange method currently being used.

Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the ASA CLI. You can usually recover access by restarting the ASA. However, if you already saved your configuration, you might be locked out. Table 41-2 lists the common lockout conditions and how you might recover from them.

Table 41-2 CLI Authentication and Command Authorization Lockout Scenarios

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users have been configured 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 into the ASA 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	The server is 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 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.	1. If the server is unreachable because the network configuration is incorrect on the ASA, session into the ASA 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 that you do not get locked out when the server is down.
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 ASA immediately, then log into the maintenance partition and reset the passwords and **aaa** commands.	Session into the ASA 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 into the ASA from the switch. From the system execution space, you can change to the context and change the user level.

Feature History for Management Access

Table 41-3 lists each feature change and the platform release in which it was implemented.

Table 41-3 Feature History for Management Access

Feature Name	Platform Releases	Feature Information
Management Access	7.0(1)	We introduced this feature. We introduced the following commands: **show running-config all privilege all, show running-config privilege level, show running-config privilege command, telnet, telnet timeout, ssh, ssh timeout, http, http server enable, asdm image disk, banner, console timeout, icmp, ipv6 icmp, management access, aaa authentication console, aaa authentication enable console, aaa authentication telnet | ssh console, service-type, login, privilege, aaa authentication exec authentication-server, aaa authentication command LOCAL, aaa accounting serial | telnet | ssh | enable console, show curpriv, aaa accounting command privilege**.
Increased SSH security; the SSH default username is no longer supported.	8.4(2)	Starting in 8.4(2), you can no longer connect to the ASA using SSH with the pix or asa username and the login password. To use SSH, you must configure AAA authentication using the **aaa authentication ssh console LOCAL** command (CLI) or Configuration > Device Management > Users/AAA > AAA Access > Authentication (ASDM); then define a local user by entering the **username** command (CLI) or choosing Configuration > Device Management > Users/AAA > User Accounts (ASDM). If you want to use a AAA server for authentication instead of the local database, we recommend also configuring local authentication as a backup method.

Table 41-3 Feature History for Management Access (continued)

Feature Name	Platform Releases	Feature Information
Support for administrator password policy when using the local database	8.4(4.1), 9.1(2)	When you configure authentication for CLI or ASDM access using the local database, you can configure a password policy that requires a user to change their password after a specified amount of time and also requires password standards such as a minimum length and the minimum number of changed characters. We introduced the following commands: **change-password, password-policy lifetime**, **password-policy minimum changes**, **password-policy minimum-length**, **password-policy minimum-lowercase**, **password-policy minimum-uppercase**, **password-policy minimum-numeric**, **password-policy minimum-special**, **password-policy authenticate enable**, **clear configure password-policy**, **show running-config password-policy**.
Support for SSH public key authentication	8.4(4.1), 9.1(2)	You can enable public key authentication for SSH connections to the ASA on a per-user basis. You can specify a public key file (PKF) formatted key or a Base64 key. The PKF key can be up to 4096 bits. Use PKF format for keys that are too large to for the ASA support of the Base64 format (up to 2048 bits). We introduced the following commands: **ssh authentication**. *PKF key format support is only in 9.1(2) and later.*
Support for Diffie-Hellman Group 14 for the SSH Key Exchange	8.4(4.1), 9.1(2)	Support for Diffie-Hellman Group 14 for SSH Key Exchange was added. Formerly, only Group 1 was supported. We introduced the following command: **ssh key-exchange**.
Support for a maximum number of management sessions	8.4(4.1), 9.1(2)	You can set the maximum number of simultaneous ASDM, SSH, and Telnet sessions. We introduced the following commands: **quota management-session**, **show running-config quota management-session**, **show quota management-session**.
For the ASASM in multiple context mode, support for Telnet and virtual console authentication from the switch.	8.5(1)	Although connecting to the ASASM from the switch in multiple context mode connects to the system execution space, you can configure authentication in the admin context to govern those connections.
AES-CTR encryption for SSH	9.1(2)	The SSH server implementation in the ASA now supports AES-CTR mode encryption.

Table 41-3 Feature History for Management Access (continued)

Feature Name	Platform Releases	Feature Information
Improved SSH rekey interval	9.1(2)	An SSH connection is rekeyed after 60 minutes of connection time or 1 GB of data traffic. We introduced the following command: **show ssh sessions detail**.
Improved one-time password authentication	9.1(5)	Administrators who have sufficient authorization privileges may enter privileged EXEC mode by entering their authentication credentials once. The **auto-enable** option was added to the **aaa authorization exec** command. We modified the following command: **aaa authorization exec**.