

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.

- Configure Management Remote Access, on page 1
- Configure AAA for System Administrators, on page 22
- Monitoring Device Access, on page 45
- History for Management Access, on page 47

Configure Management Remote Access

This section describes how to configure ASA access for ASDM, Telnet, or SSH, and other management parameters such as a login banner.

Configure SSH Access

SSH Guidelines

- To access the ASA interface for SSH access, you do not also need an access rule allowing the host IP address. You only need to configure SSH access according to this section.
- SSH access to an interface other than the one from which you entered the ASA is not supported. For example, if your SSH host is located on the outside interface, you can only initiate a management connection directly to the outside interface.
- The ASA allows a maximum of 5 concurrent SSH connections per context/single mode, with a maximum of 100 connections divided among all contexts. However, because configuration commands might obtain locks on resources being changed, you should make changes in one SSH session at a time to ensure all changes are applied correctly.
- SSH does not support:
  - EDDSA key pair
  - RSA key pair in FIPS mode
- To use the ASA **copy** command to copy a file to or from an SCP server, you have to:
  - Enable SSH access on the ASA for the SCP server subnet/host using the **ssh** command.

- Generate a key pair (for physical ASAs only) using the **crypto key generate** command.

- When using failover, if you configure the **banner motd** command, then the banner shows information about the failover state and the last failover time of the unit you are logging into. This information is useful if you are performing actions at the CLI, such as troubleshooting, and a failover occurs between sessions.

  See Configure a Login Banner, on page 19.

- 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.

- Only SSH Version 2 is supported.

To configure SSH access to the ASA, enable the SSH server and identify the allowed IP addresses. To authenticate users, you can use the following methods:

- Username and password using either the local database or a AAA server.
- Username and public key using the local database.
- X.509v3 certificate (username is derived from the certificate) and authorization from the local database or a AAA server.

By default, either the X.509 certificate or public key authentication method is attempted, depending on the SSH client's supported algorithms, but both will not be tried in the same session. If neither X.509 nor public key authentication is successful, then the ASA will attempt password authentication. You can disallow a method if needed.

Enable the SSH Server

Enable the SSH server and specify which IP addresses are allowed to connect. You can also configure other SSH server settings.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Identify the IP addresses from which the ASA accepts connections for each address or subnet, and the interface on which you can use SSH.

ssh *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface.

Unlike Telnet, you can SSH on the lowest security level interface.

Example:

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

Step 2 (Optional) Set the duration for how long an SSH session can be idle before the ASA disconnects the session.

ssh timeout *minutes*

Example:

```
ciscoasa(config)# ssh timeout 30
```

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 3 (Optional) Disallow one or more authentication methods.

ssh authentication method {[**publickey**] [**x509-certificate**] [**password**]}

By default, either the X.509 certificate or public key authentication method is attempted, depending on the SSH client's supported algorithms, but both will not be tried in the same session. If neither X.509 nor public key authentication is successful, then the ASA will attempt password authentication.

Omit the method you do not want to use. The order of the methods in the command doesn't affect the order of methods attempted.

Example:

To disallow password authentication, configure:

```
ciscoasa(config)# ssh authentication method publickey x509-certificate
```

Step 4 (Optional) Enable the Secure Copy (SCP) server.

ssh scopy enable

The SCP server does not have directory support. The lack of directory support limits remote client access to the ASA internal files.

The SCP server does not support banners or wildcards.

Step 5 (Optional) Configure SSH cipher encryption algorithms:

ssh cipher encryption {**all** | **fips** | **high** | **low** | **medium** | **custom** *colon-delimited_list_of_encryption_ciphers*}

Example:

```
ciscoasa(config)# ssh cipher encryption custom 3des-cbc:aes128-cbc:aes192-cbc
```

The default is **medium**. Ciphers are used in the order they are listed. For pre-defined lists, they are listed from highest to lowest security.

- The **all** keyword specifies using all ciphers: AEAD_AES_256_GCM aes256-gcm@openssh.com 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes128-gcm@openssh.com chacha20-poly1305@openssh.com aes192-ctr aes256-ctr aes256-gcm@openssh.com
- The **custom** keyword specifies a custom cipher encryption configuration string, separated by colons.

- The **fips** keyword specifies only FIPS-compliant ciphers: AEAD_AES_256_GCM aes256-gcm@openssh.com aes128-cbc aes256-cbc
- The **high** keyword specifies only high-strength ciphers: AEAD_AES_256_GCM aes256-gcm@openssh.com aes256-cbc aes128-gcm@openssh.com chacha20-poly1305@openssh.com aes256-ctr aes256-gcm@openssh.com
- The **low** keyword specifies low, medium, and high strength ciphers: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr AEAD_AES_256_GCM aes256-gcm@openssh.com
- The **medium** keyword specifies the medium and high strength ciphers (the default): 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr AEAD_AES_256_GCM aes256-gcm@openssh.com

Step 6 (Optional) Configure SSH cipher integrity algorithms:

ssh cipher integrity {**all** | **fips** | **high** | **low** | **medium** | **custom** *colon-delimited_list_of_integrity_ciphers*}

Example:

```
ciscoasa(config)# ssh cipher integrity custom hmac-sha1-96:hmac-md5
```

The default is **high**.

- The **all** keyword specifies using all ciphers: hmac-sha1 hmac-sha1-96 (Deprecated) hmac-md5 (Deprecated) hmac-md5-96 (Deprecated) hmac-sha2-256
- The **custom** keyword specifies a custom cipher encryption configuration string, separated by colons.
- The **fips** keyword specifies only FIPS-compliant ciphers: hmac-sha1 hmac-sha2-256
- The **high** keyword specifies only high-strength ciphers (the default): hmac-sha2-256
- The **low** keyword specifies low, medium, and high strength ciphers: hmac-sha1 hmac-sha1-96 hmac-md5 hmac-md5-96 hmac-sha2-256
- The **medium** keyword specifies the medium and high strength ciphers: hmac-sha1 hmac-sha1-96 (Deprecated) hmac-sha2-256

Step 7 (Optional) (Admin context only) Set the Diffie-Hellman (DH) key exchange mode:

ssh key-exchange group {**curve25519-sha256** | **dh-group14-sha1** | **dh-group14-sha256** | **ecdh-sha2-nistp256**}

Example:

```
ciscoasa(config)# ssh key-exchange group dh-group14-sha1
```

The default is **dh-group14-sha256**

The 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. For more information about using DH key-exchange methods, see RFC 4253. You can only set the key exchange in the Admin context; this value is used by all contexts.

Examples

The following example shows an SCP session to the ASA. From a client on the external host, perform an SCP file transfer. For example, in Linux enter the following command:

scp -v -pw *password* [*path*/]*source_filename*
*username@asa_address***:** {**disk0**|**disk1**}**:/**[*path*/]*dest_filename*

The **-v** is for verbose, and if **-pw** is not specified, you will be prompted for a password.

Configure SSH for Password Access

Configure SSH authentication using a username and password.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Generate a key pair, which is required for SSH (for physical ASAs only).

For the ASA Virtual, the key pairs are automatically created after deployment.

a) Generate the default key pair.

crypto key generate {**ecdsa elliptic-curve** *size* |**rsa modulus** *size*}

Example:

```
ciscoasa(config)# crypto key generate ecdsa elliptic-curve 521
```

- **ecdsa elliptic-curve** *size*—The size in bits is 256, 384, or 521.
- **rsa modulus** *size*—The size in bits is 2048, 3072, or 4096.

The keys used by SSH are called <Default-*type*-Key>. Do not specify the **label** keyword; non-default key pairs are not used. The larger the key size you specify, the longer it takes to generate a key pair. SSH tries keys in the following order: ECDSA, and then RSA. View the keys using the **show crypto key mypubkey** {**ecdsa** | **rsa**} command.

b) (Optional) If you do not want to use the default key order (ECDSA, and then RSA), identify the key pair you want to use.

ssh key-exchange hostkey {**rsa** | **ecdsa**}

If you choose RSA, you must use a key size 2048 or higher. For upgrade compatibility, smaller keys are only supported when you use the default key order.

Example:

```
ciscoasa(config)# ssh key-exchange hostkey ecdsa
```

Step 2 Save the keys to persistent flash memory.

write memory

Example:

```
ciscoasa(config)# write memory
```

Step 3 Create a user in the local database or on a AAA server that can be used for SSH access. See the following command to add a local username (recommended).

username *name* **password** *password* **privilege** *level*

You can use public key authentication or password authentication for the same local user. Public key authentication is not supported using a AAA server.

Example:

```
ciscoasa(config)# username admin password Far$cape1999 privilege 15
```

By default, the privilege level is 2; enter a level between 0 and 15, where 15 has all privileges.

Step 4 Enable local or AAA server authentication for SSH access.

aaa authentication ssh console {**LOCAL** | *server_group* [**LOCAL**]}

This command does not affect local public key authentication for usernames that have the **ssh authentication** command set. The ASA implicitly uses the local database for public key authentication. This command only affects local usernames with passwords. If you want to allow either public key authentication or password use by a local user, then you need to explicitly configure local authentication with this command to allow password access.

Example:

```
ciscoasa(config)# aaa authentication ssh console LOCAL
```

Configure SSH for Public Key Access

Configure SSH authentication using a public key.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Generate a key pair, which is required for SSH (for physical ASAs only).

For the ASA Virtual, the key pairs are automatically created after deployment.

a) Generate the default key pair.

crypto key generate {**ecdsa elliptic-curve** *size* |**rsa modulus** *size*}

Example:

```
ciscoasa(config)# crypto key generate ecdsa elliptic-curve 521
```

- **ecdsa elliptic-curve** *size*—The size in bits is 256, 384, or 521.
- **rsa modulus** *size*—The size in bits is 2048, 3072, or 4096.

The keys used by SSH are called <Default-*type*-Key>. Do not specify the **label** keyword; non-default key pairs are not used. The larger the key size you specify, the longer it takes to generate a key pair. SSH tries keys in the following order: ECDSA, and then RSA. View the keys using the **show crypto key mypubkey** {**ecdsa** | **rsa**} command.

b) (Optional) If you do not want to use the default key order (ECDSA, and then RSA), identify the key pair you want to use.

ssh key-exchange hostkey {**rsa** | **ecdsa**}

If you choose RSA, you must use a key size 2048 or higher. For upgrade compatibility, smaller keys are only supported when you use the default key order.

Example:

```
ciscoasa(config)# ssh key-exchange hostkey ecdsa
```

Step 2 Save the keys to persistent flash memory.

write memory

Example:

```
ciscoasa(config)# write memory
```

Step 3 Create a user in the local database that can be used for SSH access.

username *name* [**password** *password*] **privilege** *level*

Example:

```
ciscoasa(config)# username admin password Far$cape1999 privilege 15
```

By default, the privilege level is 2; enter a level between 0 and 15, where 15 has all privileges. You might want to create a user without a password if you want to force the user to use public key authentication (**ssh authentication**) instead of password authentication. If you configure public key authentication as well as a password in the **username** command, then the user can log in with either method if you explicitly configure AAA authentication in Configure SSH for Password Access, on page 5. **Note:** Do not use the **username** command **nopassword** option; the **nopassword** option allows *any* password to be entered, not no password.

Step 4 Allow public key authentication for a user instead of/as well as password authentication, and enter the public key on the ASA:

username *name* **attributes**

ssh authentication {**pkf** | **publickey** *key*}

Example:

```
ciscoasa(config)# username admin attributes
ciscoasa(config-username)# ssh authentication pkf

Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "256-bit ED25519, converted by dean@dwinchester-mac from "
AAAAC3NzaC1lZDI1NTE5AAAAIDmIeTNfEOnuH0094p1MKX80fW2O216g4trnf7gwWe5Q
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
```

For a local **username**, you can enable public key authentication instead of/as well as password authentication. You can generate a public key/private key pair using any SSH key generation software (such as ssh keygen) that can generate ssh-rsa, or ecdsa-sha2-nistp raw keys (with no certificates). Enter the public key on the ASA. The SSH client then uses the private key (and the passphrase you used to create the key pair) to connect to the ASA.

For a **pkf** key, you are prompted to paste in a PKF formatted key, up to 4096 bits. Use this format for keys that are too large to paste inline in Base64 format. For example, you can generate a 4096-bit key using ssh keygen, then convert it to PKF, and use the **pkf** keyword to be prompted for the key. **Note:** You can use the **pkf** option with failover, but the PKF key is not automatically replicated to the standby system. You must enter the **write standby** command to synchronize the PKF key.

For a **publickey** *key*, the key is a Base64-encoded public key. You can generate the key using any SSH key generation software (such as ssh keygen) that can generate ssh-rsa, or ecdsa-sha2-nistp raw keys (with no certificates).

Examples

The following example shows how to authenticate using a PKF formatted key:

```
ciscoasa(config)# crypto key generate rsa modulus 4096
ciscoasa(config)# write memory
ciscoasa(config)# username exampleuser1 password examplepassword1 privilege 15
ciscoasa(config)# username exampleuser1 attributes
ciscoasa(config-username)# ssh authentication pkf
Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by xxx@xxx from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnFaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77II==
```

```
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
ciscoasa(config)#
```

The following example generates a shared key for SSH on a Linux or Macintosh system, and imports it to the ASA:

1. Generate the RSA public and private keys for 4096 bits on your computer:

```
jcrichton-mac:~ john$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/john/.ssh/id_rsa):
/Users/john/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): pa$$phrase
Enter same passphrase again: pa$$phrase
Your identification has been saved in /Users/john/.ssh/id_rsa.
Your public key has been saved in /Users/john/.ssh/id_rsa.pub.
The key fingerprint is:
c0:0a:a2:3c:99:fc:00:62:f1:ee:fa:f8:ef:70:c1:f9 john@jcrichton-mac
The key's randomart image is:
+--[ RSA 4096]----+
| .               |
|  o  .           |
|+...  o          |
|B.+.....         |
|.B ..+  S        |
|  =   o          |
|   + . E         |
|  o o            |
| ooooo           |
+-----------------+
```

2. Convert the key to PKF format:

```
jcrichton-mac:~ john$ cd .ssh
jcrichton-mac:.ssh john$ ssh-keygen -e -f id_rsa.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by john@jcrichton-mac from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnDaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77IQ==
---- END SSH2 PUBLIC KEY ----
jcrichton-mac:.ssh john$
```

3. Copy the key to your clipboard.
4. In ASDM, choose **Configuration** > **Device Management** > **Users/AAA** > **User Accounts**, select the username and then click **Edit**. Click **Public Key Using PKF** and paste the key into the window:

5. Verify the user can SSH to the ASA. For the password, enter the SSH key password you specified when you created the key pair.

```
jcrichton-mac:.ssh john$ ssh test@10.86.118.5
The authenticity of host '10.86.118.5 (10.86.118.5)' can't be established.
RSA key fingerprint is 39:ca:ed:a8:75:5b:cc:8e:e2:1d:96:2b:93:b5:69:94.
Are you sure you want to continue connecting (yes/no)? yes
```

The following dialog box appears for you to enter your passphrase:



Meanwhile, in the terminal session:

```
Warning: Permanently added '10.86.118.5' (RSA) to the list of known hosts.
Identity added: /Users/john/.ssh/id_rsa (/Users/john/.ssh/id_rsa)
Type help or '?' for a list of available commands.
asa>
```

Configure SSH for X.509 Certificate Access

Enable X.509v3 certificate authentication and authorization.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Configure the trustpoint. See Configure Digital Certificates.

Step 2 Enable the trustpoint to validate SSH clients.

crypto ca trustpoint *trustpoint_name*

validation-usage ssh-client [**ipsec-client**] [**ssl-client**] [**ssl-server**]

You can also use this trustpoint or IPsec clients, SSL clients, and SSL servers if you specify these additional types.

Example:

```
ciscoasa(config)# crypto ca trustpoint lets-encrypt_CA
ciscoasa(config-ca-trustpoint)# validation-usage ssh-client ipsec-client ssl-client ssl-server
```

Step 3 Add a user to the authorization server that matches the username derived from the certificate.

- AAA server—See AAA Servers and Server Groups to add a AAA server group to the ASA.
- Local database—See Add a User Account to the Local Database.

If you want to allow password authentication for this user as well, you can configure the user password. If you do not configure password authentication, you can add the user without a password. In this case, if you set a password, it will be ignored.

Console privileges are assigned based on the attributes returned from the authorization server. See Control CLI and ASDM Access with Management Authorization, on page 29 for attributes to set for the user.

Example:

```
ciscoasa(config)# username dwinchester attributes
ciscoasa(config-username)# service-type admin
```

Step 4 Identify the trustpoint to be used for SSH.

ssh trustpoint sign *trustpoint_name*

Example:

```
ciscoasa(config)# ssh trustpoint sign ?

configure mode commands/options:
Available configured trustpoints:
  _SmartCallHome_ServerCA
  lets-encrypt_CA

ciscoasa(config)# ssh trustpoint sign lets-encrypt_CA
```

Step 5 Define how the ASA derives the username from the certificate.

ssh username-from-certificate { [**cn**] [**upn**]}

Specify one or both options. The order of the options in the command determines the order tried.

- **cn**—Uses the common-name.
- **upn**—Uses the user-principle-name.

Example:

```
ciscoasa(config)# ssh username-from-certificate cn
```

Step 6 Authorize the user using the local database or a AAA server.

aaa authorization exec ssh-x509 {*server_group* | **LOCAL**} [**auto-enable**]

- **auto-enable**—Lets administrators who have sufficient authorization privileges enter privileged EXEC mode automatically when they log in.

```
ciscoasa(config)# aaa authorization exec ssh-x509 LOCAL auto-enable
```

Example

The following example enables SSH X.509 authentication with authorization using the local database:

```
ciscoasa(config)# crypto ca trustpoint lets-encrypt_CA
ciscoasa(config-ca-trustpoint)# validation-usage ssh-client ipsec-client ssl-client ssl-server
ciscoasa(config-ca-trustpoint)# exit
ciscoasa(config)# username dwinchester attributes
ciscoasa(config-username)# service-type admin
ciscoasa(config-username)# exit
ciscoasa(config)# aaa authorization exec ssh-x509 LOCAL auto-enable
ciscoasa(config)# ssh trustpoint sign lets-encrypt_CA
ciscoasa(config)# ssh username-from-certificate cn upn
```

Configure Telnet Access

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

- To access the ASA interface for Telnet access, you do not also need an access rule allowing the host IP address. You only need to configure Telnet access according to this section.
- Telnet access to an interface other than the one from which you entered the ASA is not supported. For example, if your Telnet host is located on the outside interface, you can only initiate a Telnet connection directly to the outside interface. The only exception to this rule is through a VPN connection. See Configure Management Access Over a VPN Tunnel, on page 16.
- You cannot use Telnet to the lowest security interface unless you use Telnet inside a VPN tunnel.
- The ASA allows a maximum of 5 concurrent Telnet connections per context/single mode, with a maximum of 100 connections divided among all contexts.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.
- To gain access to the ASA CLI using Telnet, enter the login password set by the **password** command. You must manually set the password before using Telnet.

Procedure

Step 1 Identify the IP addresses from which the ASA accepts connections for each address or subnet on the specified interface.

telnet *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface. For VPN management access only (see Configure Management Access Over a VPN Tunnel, on page 16), specify the named BVI 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.

Example:

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

Step 2 Set the duration for how long a Telnet session can be idle before the ASA disconnects the session.

telnet timeout *minutes*

Example:

```
ciscoasa(config)#  telnet timeout 30
```

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. 255.255.255.255 inside
```

Configure HTTPS Access for ASDM, Other Clients

To use ASDM or other HTTPS clients such as CSM, 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. To configure HTTPS access, perform the following steps. See the following guidelines:

- To access the ASA interface for HTTPS access, you do not also need an access rule allowing the host IP address. You only need to configure HTTPS access according to this section. If, however, you configure HTTP redirect to redirect HTTP connections to HTTPS automatically, you must enable an access rule to allow HTTP; otherwise, the interface cannot listen to the HTTP port.
- 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 Configure Management Access Over a VPN Tunnel, on page 16.

- In single context mode, you can have a maximum 5 ASDM concurrent sessions. In multiple context mode, you can have a maximum of 5 concurrent ASDM sessions per context, with a maximum of 200 ASDM instances among all contexts.

  ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the multiple-context mode system limit of 200 ASDM sessions represents a limit of 400 HTTPS sessions.

- The ASA allows a maximum of 6 concurrent non-ASDM HTTPS sessions in single context mode or per context, if available, with a maximum or 100 HTTPS sessions among all contexts.

- If you enable both SSL (**webvpn** > **enable** *interface*) and HTTPS access on the same interface, you can access Secure Client from **https://***ip_address* and ASDM from **https://***ip_address***/admin**, both on port 443. If you also enable **aaa authentication http console**, then you must specify a different port for ASDM access.

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter **changeto context** *name*.

Procedure

Step 1 Identify the IP addresses from which the ASA accepts HTTPS connections for each address or subnet on the specified interface.

http *source_IP_address mask source_interface*

- *source_interface*—Specify any named interface. For bridge groups, specify the bridge group member interface. For VPN management access only (see Configure Management Access Over a VPN Tunnel, on page 16), specify the named BVI interface.

Example:

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

Step 2 Enable the HTTPS server.

http server enable [*port*]

Example:

```
ciscoasa(config)# http server enable 444
```

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 URL:

https://10.1.1.1:444

Step 3 Allow non-browser-based HTTPS clients to access HTTPS services on the ASA. By default, ASDM, CSM, and REST API are allowed.

http server basic-auth-client *user_agent*

- *user_agent*—Specifies the client's User-Agent string in the HTTP header of the HTTP request. You can specify the complete string or a partial string; partial strings must match the start of the User-Agent string. We recommend complete strings for better security. Note that the string is case-sensitive.

  For example, **curl** will match the following User-Agent string:

  ```
  curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic
  ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  ```

  curl will *not* match the following User-Agent string:

  ```
  abcd curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1
  Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  ```

  CURL will *not* match the following User-Agent string:

  ```
  curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic
  ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  ```

Enter each client string using a separate command. Many specialty clients (for example, python libraries, curl, and wget) do not support Cross-site request forgery (CSRF) token-based authentication, so you need to specifically allow these clients to use the ASA basic authentication method. For security purposes, you should only allow required clients.

Example:

```
ciscoasa(config)# http server basic-auth-client curl
```

Step 4 (Optional) Set connection and session timeouts.

http server idle-timeout*minutes*

http server session-timeout*minutes*

http connection idle-timeout*seconds*

- **http server idle-timeout** *minutes*—Set the idle timeout for ASDM connections, from 1-1440 minutes. The default is 20 minutes. The ASA disconnects an ASDM connection that is idle for the set period of time.
- **http server session-timeout** *minutes*—Set the session timeout for ASDM sessions, from 1-1440 minutes. This timeout is disabled by default. The ASA disconnects an ASDM session that exceeds the set period of time.
- **http connection idle-timeout** *seconds*—Set the idle timeout for all HTTPS connections, including ASDM, WebVPN, and other clients, from 10-86400 seconds. This timeout is disabled by default. The ASA disconnects a connection that is idle for the set period of time. If you set both the **http server idle-timeout** and the **http connection idle-timeout** commands, the **http connection idle-timeout** command takes precendence.

Example:

```
ciscoasa(config)# http server idle-timeout 30
```

```
ciscoasa(config)# http server session-timeout 120
```

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 192.168.3.0/24 network to access ASDM on the inside interface:

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

Configure HTTP Redirect for ASDM Access or Clientless SSL VPN

You must use HTTPS to connect to the ASA using ASDM or clientless SSL VPN. For your convenience, you can redirect HTTP management connections to HTTPS. For example, by redirecting HTTP, you can enter either **http://10.1.8.4/admin/** or **https://10.1.8.4/admin/** and still arrive at the ASDM launch page at the HTTPS address.

You can redirect both IPv4 and IPv6 traffic.

Before you begin

Normally, you do not need an access rule allowing the host IP address. However, for HTTP redirect, you must enable an access rule to allow HTTP; otherwise, the interface cannot listen to the HTTP port.

Procedure

Enable HTTP redirect:

http redirect *interface_name* [*port*]

Example:

```
ciscoasa(config)# http redirect outside 88
```

The *port* identifies the port from which the interface redirects HTTP connections. The default is 80.

Configure 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 must 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 or Telnet; or you can ping the inside interface when entering from the outside interface.

Note This feature is not supported for SSH.

Note This feature is not supported for SNMP. For SNMP over VPN, we recommend enabling SNMP on a loopback interface. You don't need the management-access feature enabled to use SNMP on the loopback interface. Loopback also works for SSH.

VPN access to an interface other than the one from which you entered the ASA is not supported. For example, if your VPN access is located on the outside interface, you can only initiate a connection directly to the outside interface. You should enable VPN on the directly-accessible interface of the ASA and use name resolution so that you don’t have to remember multiple addresses.

Management access is available via the following VPN tunnel types: IPsec clients, IPsec Site-to-Site, Easy VPN, and the Secure Client SSL VPN.

Before you begin

- This feature is not supported on management-only interfaces.
- When you use a management-access interface and you configure identity NAT, you must configure NAT with the route lookup option. For more information see the "NAT and VPN Management Access" section in the *NAT Examples and Reference* chapter in the appropriate release of the ASA Firewall CLI Configuration Guide.

Procedure

Specify the name of the management interface that you want to access when entering the ASA from another interface.

management-access *management_interface*

For site-to-site tunnels, you can specify a named BVI (in routed mode).

Example:

```
ciscoasa(config)# management-access inside
```

Change 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.

Procedure

Specify the idle time in minutes (0 through 60) after which the privileged session ends.

console timeout *number*

Example:

```
ciscoasa(config)# console timeout 0
```

The default timeout is 0, which means the session does not time out.

Customize a CLI Prompt

The ability to add information to a prompt allows you to see at-a-glance which ASA you are logged into when you have multiple modules. During a failover, this feature is useful when both ASAs have the same hostname.

In multiple context mode, you can view the extended prompt when you log in to the system execution space or the admin context. Within a non-admin context, you only see the default prompt, which is the hostname and the context name.

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	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 or role of the unit. For failover, the following values are displayed for the state keyword: • **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 other non-active 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 might happen when there is an interface failure above the threshold on the standby unit. For clustering, the values for control and data are shown.

Procedure

Customize the CLI prompt by entering the following command:

prompt {[**hostname**] [**context**] [**domain**] [**slot**] [**state**] [**priority**] [**cluster-unit**]}

Example:

```
ciscoasa(config)# prompt hostname context slot state priority
ciscoasa/admin/pri/act(config)#
```

The order in which you enter the keywords determines the order of the elements in the prompt, which are separated by a slash (/).

Configure 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.

- 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.
```

- After a banner has been added, Telnet or SSH sessions to the 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).

- When using failover, if you configure the **banner motd** command, then the banner shows information about the failover state and the last failover time of the unit you are logging into. This information is useful if you are performing actions at the CLI, such as troubleshooting, and a failover occurs between sessions.
- See RFC 2196 for guidelines about banner messages.

Procedure

Add 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).

banner {**exec** | **login** | **motd**} *text*

Example:

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).
```

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 examples show how to add a message-of-the-day banner:

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).

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

The following example shows the information displayed on login to the active unit in a failover pair when the **banner motd** command is configured.

```
Last login: Fri Jun 27 06:15:29 UTC 2025 from 10.16.0.101 on pts/0
Successful login attempts for user 'dwinchester' : 1
Last login: Fri Jun 27 06:16:21 2025 from 10.16.0.101

               State          Last Failure Reason      Date/Time
```

```
This host  -   Primary
               Active         None
Other host -   Secondary
               Standby Ready  Comm Failure             05:40:48 UTC Jun 27 2025

====Configuration State===
 Sync Done
====Communication State===
 Mac set

Last Failover at: 05:40:30 UTC Jun 27 2025
"This is the active device"
User dwinchester logged in to active
Logins over the last 1 days: 3.  Last login: 06:15:29 UTC Jun 27 2025 from console
Failed logins since the last login: 0.
Type help or '?' for a list of available commands.
active>
```

Set a Management Session Quota

You can establish a maximum number of simultaneous ASDM, SSH, and Telnet sessions that are allowed on the ASA. 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.

Note In multiple context mode, you cannot configure the number of ASDM sessions, where the maximum is fixed at 5 sessions.

Note If you also set a resource limit per context for the maximum administrative sessions (SSH, etc.), then the lower value will be used.

Before you begin

In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name*command.

Procedure

Step 1 Enter the following command:

quota management-session [**ssh** | **telnet** | **http** | **user**] *number*

- **ssh**—Sets the maximum SSH sessions, between 1 and 5. The default is 5.
- **telnet**—Sets the maximum Telnet sessions, between 1 and 5. The default is 5.
- **http**—Sets the maximum HTTPS (ASDM) sessions, between 1 and 5. The default is 5.
- **user**—Sets the maximum sessions per user, between 1 and 5. The default is 5.

- *number*—Sets the number of sessions. When entered without any other keywords, this argument sets the aggregate number of sessions between 1 and 15. The default is 15.

Example:

```
ciscoasa(config)# quota management-session ssh 3
ciscoasa(config)# quota management-session telnet 1
ciscoasa(config)# quota management-session http 4
ciscoasa(config)# quota management-session user 2
```

Step 2 View the current sessions in use.

show quota management-session [**ssh** | **telnet** | **http** | **user**]

Example:

```
ciscoasa(config)#show quota management-session

#Sessions               ConnectionType                 Username
1                              SSH                     cisco
2                              TELNET                  cisco
1                              SSH                     cisco1
```

Configure AAA for System Administrators

This section describes how to configure authentication, management authorization, and command authorization for system administrators.

Configure Management Authentication

Configure authentication for CLI and ASDM access.

About Management Authentication

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

About SSH Authentication

See the following behavior for SSH access with and without authentication:

- No Authentication—SSH is not available without authentication.
- Authentication—When you enable SSH authentication, you enter the username and password as defined on the AAA server or local user database. For public key authentication, the ASA only supports the local database. If you configure SSH public key authentication, then the ASA uses the local database implicitly. You only need to explicitly configure SSH authentication when you use a username and password to log in. You access user EXEC mode.

About Telnet Authentication

See the following behavior for Telnet access with and without 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). There is no default password, so you must set one before you can Telnet to the ASA. You access user EXEC mode.
- Authentication—If you enable Telnet authentication, you enter the username and password as defined on the AAA server or local user database. You access user EXEC mode.

About ASDM Authentication

See the following behavior for ASDM access with and without authentication. You can also configure certificate authentication, with or without AAA authentication.

- No Authentication—By default, you can log into ASDM with a blank username and the enable password set by the **enable password** command, which is blank by default. We suggest that you change the enable password as soon as possible so that it does not remain blank; see Set the Hostname, Domain Name, and the Enable and Telnet Passwords. When you enter the **enable** command at the CLI for the first time, you are prompted to change the password; this behavior is not enforced when you log into ASDM. 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.
- Certificate Authentication—(Single, routed mode only) You can require that the user have a valid certificate. Enter the certificate username and password, and the ASA validates the certificate against the PKI trustpoint.
- AAA Authentication—When you enable ASDM (HTTPS) authentication, you enter the username and password as defined on the AAA server or local user database. You can no longer use ASDM with a blank username and the enable password.
- AAA Authentication plus Certificate Authentication—(Single, routed mode only) When you enable ASDM (HTTPS) authentication, you enter the username and password as defined on the AAA server or local user database. If the username and password are different for the certificate authentication, you are prompted to enter them as well. You can opt to pre-fill the username derived from your certificate.

About Serial Authentication

See the following behavior for access to the serial console port with and without authentication:

- No Authentication—If you do not enable any authentication for serial access, you do not enter a username or password. You access user EXEC mode.
- Authentication—If you enable authentication for serial access, you enter the username and password as defined on the AAA server or local user database. You access user EXEC mode.

About Enable Authentication

To enter privileged EXEC mode after logging in, enter the **enable** command. How this command works depends on whether or not 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), which is blank by default. The first time you enter the **enable** command, you are prompted to change it. However, if you do not use enable authentication, after you enter the **enable** command, you are no longer logged in as a particular

user, which can affect user-based features such as command authorization. To maintain your username, use enable authentication.

- Authentication—If you configure enable authentication, the ASA prompts you for your username and password as defined on the AAA server or local user database. 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. The **login** command maintains the username, but requires no configuration to turn on authentication.



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 discourage the login command by using a AAA server for authentication instead of the local database, 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.

Sessions from the Host Operating System to the ASA

Some platforms support running the ASA as a separate application: for example the ASA on the Firepower 4100/9300. For sessions from the host operating system to the ASA, you can configure serial and Telnet authentication, depending on the type of connection.

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 these sessions. The admin context AAA server or local user database is used in this instance.

Configure Authentication for CLI and ASDM Access

Before you begin

- Configure Telnet, SSH, or HTTP access.
- For external authentication, configure a AAA server group. For local authentication, add users to the local database.
- HTTP management authentication does not support the SDI protocol for a AAA server group.
- This feature does not affect SSH public key authentication for local usernames with the **ssh authentication** command. The ASA implicitly uses the local database for public key authentication. This feature only affects usernames with passwords. If you want to allow either public key authentication or password use by a local user, then you need to explicitly configure local authentication with this procedure to allow password access.

Procedure

Authenticate users for management access.

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
```

The **telnet** keyword controls Telnet access. The **ssh** keyword controls SSH access (password only; public key authentication implicitly uses the local database). The **http** keyword controls ASDM access. The **serial** keyword controls console port access.

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** (which 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.

Configure Enable Authentication (Privileged EXEC Mode)

You can authentication users when they enter the **enable** command.

Before you begin

See About Enable Authentication, on page 23.

Procedure

Choose one of the following options for authenticating users:

- To authenticate users with a AAA server or the local database, enter the following command:

  aaa authentication enable console {**LOCAL** | *server_group* [**LOCAL**]}

  Example:

  ```
  ciscoasa(config)# aaa authentication enable console LOCAL
  ```

  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** (which 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.

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

  login

Example:

```
ciscoasa# login
```

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.

Users can 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.

Configure ASDM Certificate Authentication

You can require certificate authentication, with or without AAA authentication. The ASA validates the certificate against the PKI trustpoint.

Before you begin

- Supported in single, routed mode only.
- You need to have a client (.pfx format) and server certificate (.p12 format). Remember the decryption passphrase for importing the certificate.

Procedure

Step 1 Add an identity certificate to the ASA. See Configure Trustpoints

Step 2 Enable certificate authentication:

http authentication-certificate *interface_name*[**match** *certificate_map_name*]

Example:

```
ciscoasa(config)# crypto ca certificate map map1 10
ciscoasa(config-ca-cert-map)# subject-name emailAddress www.example.com
ciscoasa(config)# http authentication-certificate outside match map1
```

You configure certificate authentication for each interface, so that connections on a trusted/inside interface do not have to provide a certificate. You can use the command multiple times to enable certificate authentication on multiple interfaces.

To require the certificate to match a certificate map, specify the **match** keyword and the map name. Configure the map using the **crypto ca certificate map** command.

Step 3 (Optional) Set the attribute used by ASDM to derive the username from the certificate:

http username-from-certificate{*primary-attr* [*secondary-attr*] | **use-entire-name** | **use-script**} [**pre-fill-username**]

Example:

```
ciscoasa(config)# http username-from-certificate CN pre-fill-username
```

By default, ASDM uses CN OU attributes.

- The *primary-attr* argument specifies the attribute to be used to derive the username. The *secondary-attr* argument specifies an additional attribute to use with the primary attribute to derive the username. You can use the following attributes:
  - C—Country
  - CN—Common Name
  - DNQ—DN qualifier
  - emailAddress—Email Address
  - GENQ—Generational qualifier
  - GN—Given Name
  - I—Initials
  - L—Locality
  - N—Name
  - O—Organization
  - OU—Organizational Unit
  - SER—Serial Number
  - SN—Surname
  - SP—State/Province
  - T—Title
  - UID User ID
  - UPN—User Principal Name
- The **use-entire-name** keyword uses the entire DN name.
- The **use-script** keyword uses a Lua script generated by ASDM.
- The **pre-fill-username** keyword pre-fills the username when prompted for authentication. If the username is different from the one you initially typed in, a new dialog box appears with the username pre-filled. You can then enter the password for authentication.

Step 4 Windows ASDM Launcher steps.

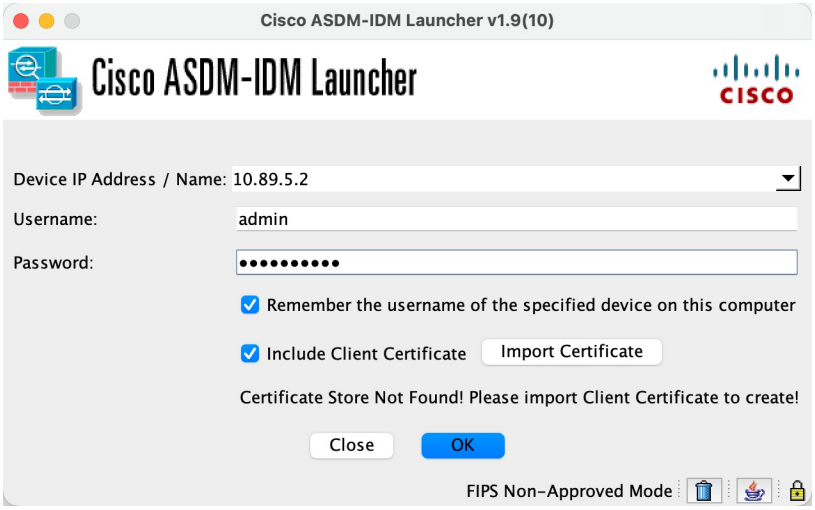
a) (FIPS Non-Approved Mode) Install the client certificate.

1. Double click the .pfx file to install the certificate.
2. Enter the decryption password.

   For more information about certificates in Windows, see https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/working-with-certificates.

b) Open the ASDM Launcher.

Figure 1: ASDM Launcher



c) Enter the **Device IP Address**, **Username**, and **Password**.

d) Check **Include Client Certificate**.

e) (FIPS Approved Mode) Click **Import Certificate** and browse to your client certificate in .pfx format. Enter the password and click **OK**.

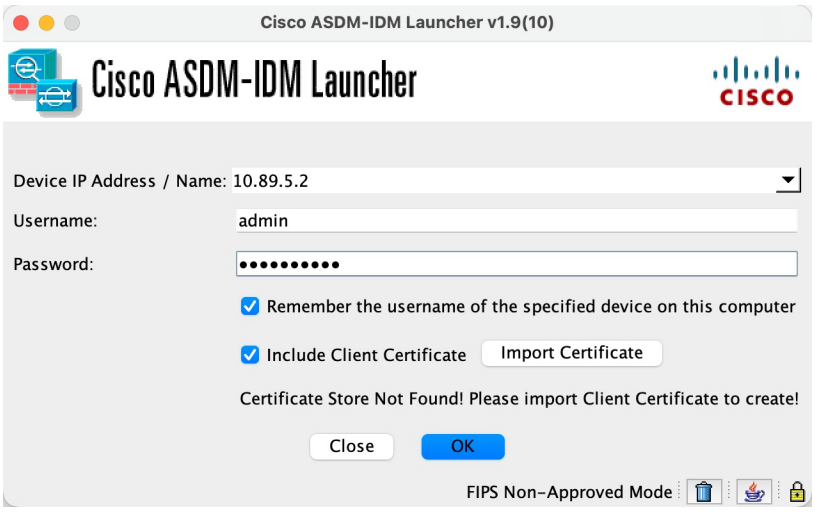
In FIPS Non-Approved Mode on Windows, the client certificate will be taken from the Windows certificate store, and you will not see the **Import Certificate** button.

f) (FIPS Approved Mode) To update a certificate, click the **Update Certificate** button and import a new certificate.

Step 5 MacOS ASDM Launcher steps.

a) Open the ASDM Launcher.

Figure 2: ASDM Launcher



b) Enter the **Device IP Address**, **Username**, and **Password**.

c) Check **Include Client Certificate**.
d) Click **Import Certificate** and browse to your client certificate in .pfx format. Enter the password and click **OK**.
e) To update a certificate, click the **Update Certificate** button and import a new certificate.

Control CLI and ASDM Access with Management Authorization

The ASA lets you distinguish between administrative and remote-access users when they authenticate. User role differentiation can prevent remote access VPN and network access users from establishing an administrative connection to the ASA.

For SSH X.509 certificate authorization, see Configure SSH for X.509 Certificate Access, on page 10.

Before you begin

RADIUS or LDAP (mapped) users

When users are authenticated through LDAP, the native LDAP attributes and their values can be mapped to 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.

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

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.

TACACS+ users

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

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.

Management Authorization Attributes

See the following table for AAA server types and valid values for management authorization. The ASA uses these values to determine the level of management access.

Management Level	RADIUS/LDAP (Mapped) Attributes	TACACS+ Attributes	Local Database Attributes
Full Access—Allows full access to any services specified by the **aaa authentication console** commands	Service-Type 6 (Administrative), Privilege-Level 1	PASS, privilege level 1	admin

Management Level	RADIUS/LDAP (Mapped) Attributes	TACACS+ Attributes	Local Database Attributes
Partial Access—Allows access to the CLI or ASDM when you configure the **aaa authentication console** commands. However, if you configure **enable** authentication with the **aaa authentication enable console** command, then the CLI user cannot access privileged EXEC mode using the **enable** command.	Service-Type 7 (NAS prompt), Privilege-Level 2 and higher The Framed (2) and Login (1) service types are treated the same way.	PASS, privilege level 2 and higher	nas-prompt
No Access—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.	Service-Type 5 (Outbound)	FAIL	remote-access

Additional Guidelines

- Serial console access is not included in management authorization.
- You must also configure AAA authentication for management access to use this feature. See Configure Authentication for CLI and ASDM Access, on page 24.
- If you use external authentication, you must pre-configure a AAA server group before you enable this feature.
- HTTP authorization is supported in single, routed mode only.

Procedure

Step 1 Enable management authorization for Telnet and SSH:

aaa authorization exec {**authentication-server** | **LOCAL**} [**auto-enable**]

The **auto-enable** keyword lets administrators who have sufficient authorization privileges enter privileged EXEC mode automatically when they log in.

Example:

```
ciscoasa(config)# aaa authentication ssh console RADIUS
ciscoasa(config)# aaa authorization exec authentication-server auto-enable
```

Step 2 Enable management authorization for HTTPS (ASDM):

aaa authorization http console {**authentication-server** | **LOCAL**}

Example:

```
ciscoasa(config)# aaa authentication http console RADIUS
ciscoasa(config)# aaa authorization http console authentication-server
```

Step 3

Examples

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
```

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
```

Configure 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

About Command Authorization

You can enable command authorization so only authorized users can enter commands.

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.

- 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.

Security Contexts and Command Authorization

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.

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

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 cannot enter configuration mode.

Configure 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).

Procedure

Step 1 Assign a command to a privilege level.

privilege [**show** | **clear** | **cmd**] **level** *level* [**mode** {**enable** | **cmd**}] **command** *command*

Example:

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

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.

Step 2 (Optional) Enable AAA users for command authorization. Without this command, the ASA only supports privilege levels for local database users and defaults all other types of users to level 15.

aaa authorization exec authentication-server [auto-enable]

Example:

```
ciscoasa(config)# aaa authorization exec authentication-server
```

This command also enables management authorization. See Control CLI and ASDM Access with Management Authorization, on page 29.

Step 3 Enable the use of local command privilege levels:

aaa authorization command LOCAL

Example:

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

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.

Configure 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.

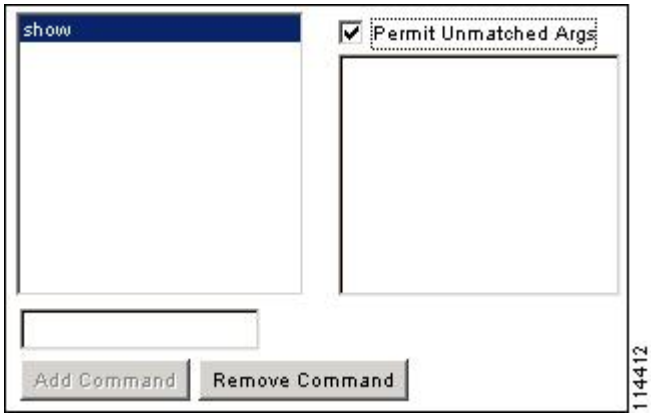
- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

  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.

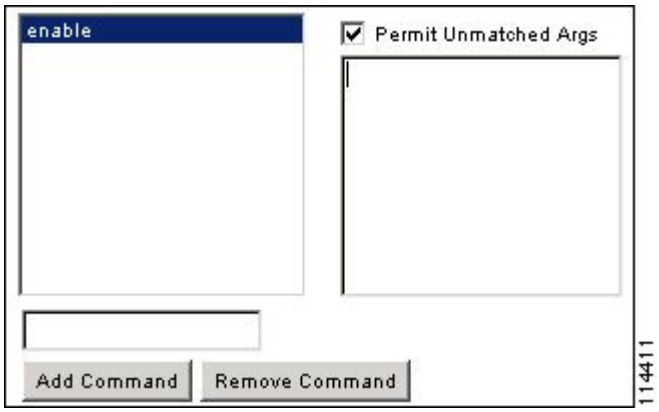
  For example, you can configure just the **show** command, then all the **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and a question mark, which shows CLI usage (see the following figure).

Figure 3: Permitting All Related Commands



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

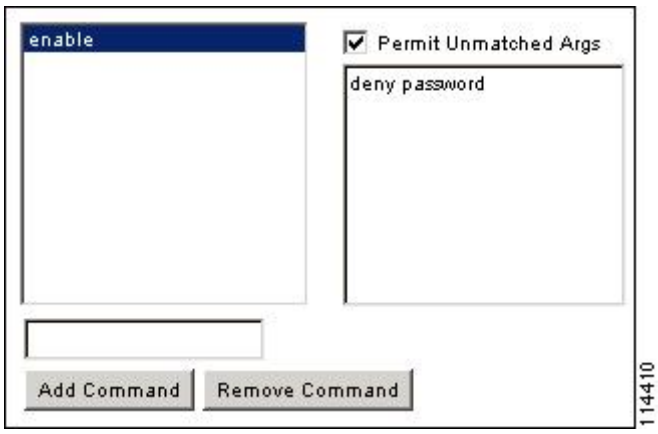
Figure 4: Permitting Single Word Commands



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

  For example, to allow **enable**, but not **enable password**, enter **enable** in the commands 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 the following figure).

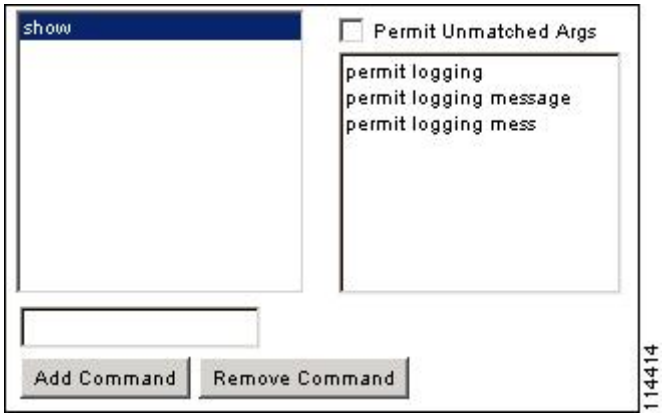
Figure 5: Disallowing Arguments



- 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 (see the following figure).

Figure 6: Specifying Abbreviations



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

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

Configure 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.

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.

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

Procedure

Enter the following command:

aaa authorization command *tacacs+_server_group* [**LOCAL**]

Example:

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

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 of which method is being used. Be sure to configure users in the local database and command privilege levels.

Configure 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.

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.

Before you begin

- Configure AAA authentication for CLI or ASDM access using the local database.
- Specify usernames in the local database.

Procedure

Step 1 (Optional) Set the interval in days after which passwords expire for remote users.

password-policy lifetime *days*

Example:

```
ciscoasa(config)# password-policy lifetime 180
```

Note Users at the console port are never locked out because of password expiration.

Valid values are between 0 and 65536 days. The default value is 0 days, a value indicating that passwords will never expire.

Seven 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 (Optional) Set the minimum number of characters that you must change between new and old passwords.

password-policy minimum-changes *value*

Example:

```
ciscoasa(config)# password-policy minimum-changes 2
```

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 (Optional) Set the minimum length of passwords.

password-policy minimum-length *value*

Example:

```
ciscoasa(config)# password-policy minimum-length 8
```

Valid values are between 3 and 64 characters. We recommend a minimum password length of 8 characters.

Step 4 (Optional) Set the minimum number of upper case characters that passwords must have.

password-policy minimum-uppercase *value*

Example:

```
ciscoasa(config)# password-policy  minimum-uppercase 3
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 5 (Optional) Set the minimum number of lower case characters that passwords must have.

password-policy minimum-lowercase *value*

Example:

```
ciscoasa(config)# password-policy minimum-lowercase 6
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 6 (Optional) Set the minimum number of numeric characters that passwords must have.

password-policy minimum-numeric *value*

Example:

```
ciscoasa(config)# password-policy minimum-numeric 1
```

Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.

Step 7 (Optional) Set the minimum number of special characters that passwords must have.

password-policy minimum-special *value*

Example:

```
ciscoasa(config)# password-policy minimum-special 2
```

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 Prohibit the reuse of a password:

password-policy reuse-interval *value*

Example:

```
ciscoasa(config)# password-policy reuse-interval 5
```

You can prohibit the reuse of a password that matches previously used passwords, between 2 and 7 previous passwords. The previous passwords are stored in the configuration under each username in encrypted form using the **password-history** command; this command is not user-configurable.

Step 9 Prohibit a password that matches a username:

password-policy username-check

Step 10 (Optional) Set whether users must change their password using the **change-password** command, instead of letting users change their password with the **username** command.

password-policy authenticate enable

Example:

```
ciscoasa(config)# password-policy authenticate enable
```

The default setting is disabled: a user can use either method to change their password.

If you enable this feature and 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
```

Change Your Password

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

To change your username password, perform the following steps:

Procedure

Enter the following command:

change-password [**old-password** *old_password* [**new-password** *new_password*]]

Example:

```
ciscoasa# change-password old-password j0hncr1chton new-password a3rynsun
```

If you do not enter the old and new passwords in the command, the ASA prompts you for input.

Enable and View the Login History

By default, the login history is saved for 90 days. You can disable this feature or change the duration, up to 365 days.

Before you begin

- The login history is only saved per unit; in failover and clustering environments, each unit maintains its own login history only.
- Login history data is not maintained over reloads.
- This feature applies to usernames in the local database or from a AAA server when you enable local AAA authentication for one or more of the CLI management methods (SSH, Telnet, serial console). ASDM logins are not saved in the history.

Procedure

Step 1 Set the login history duration:

aaa authentication login-history duration *days*

Example:

```
ciscoasa(config)# aaa authentication login-history duration 365
```

You can set the *days* between 1 and 365. The default is 90. To disable the login history, enter **no aaa authentication login-history**.

When a user logs in, they see their own login history, such as this SSH example:

```
cugel@10.86.194.108's password:
The privilege level for user cugel is 15.  The privilege level at the previous login was
2.
User cugel logged in to ciscoasa at 21:04:10 UTC Dec 14 2016
Last login: 21:01:44 UTC Dec 14 2016 from ciscoasa console
Successful logins over the last 90 days: 6
Authentication failures since the last login: 0
Type help or '?' for a list of available commands.
ciscoasa>
```

Step 2 View the login history:

show aaa login-history [**user** *name*]

Example:

```
ciscoasa(config)# show aaa login-history
Login history for user:    turjan
Logins in last   1 days:   1
Last successful login:     16:44:32 UTC Jul 23 2018 from console
Failures since last login: 0
Last failed login:         None
Privilege level:                                14
Privilege level changed from 11 to 14 at:       14:07:30 UTC Aug 21 2018
```

Configure 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:

Procedure

Step 1 Enter the following command:

aaa accounting {**serial** | **telnet** | **ssh** | **enable**} **console** *server-tag*

Example:

```
ciscoasa(config)# aaa accounting telnet console group_1
```

Valid server group protocols are RADIUS and TACACS+.

Step 2 Enable command accounting. Only TACACS+ servers support command accounting.

aaa accounting command [**privilege** *level*] *server-tag*

Example:

```
ciscoasa(config)# aaa accounting command privilege 15 group_1
```

The **privilege** *level* keyword-argument pair is the minimum privilege level and the *server-tag* argument is the name of the TACACS+ server group to which the ASA should send command accounting messages.

Recover 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.

The following table lists the common lockout conditions and how you might recover from them.

Table 1: 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.

Monitoring Device Access

See the following commands for monitoring device access:

- **show running-config all privilege all**

  This command shows privilege levels for all commands.

  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
...
```

- **show running-config privilege level** *level*

  This command shows commands for a specific privilege level. The level argument is an integer between 0 and 15.

  The following example shows the command assignments for privilege level 10:

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

- **show running-config privilege command** *command*

  This command shows the privilege level of a specific command.

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

```
ciscoasa(config)# show running-config all 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
```

- **show curpriv**

  This command shows the currently logged-in user.

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

```
ciscoasa# show curpriv
```

```
Username: admin
Current privilege level: 15
Current Mode/s: P_PRIV
```

The following table describes the **show curpriv** command output.

Table 2: 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

- **show quota management-session** [**ssh** | **telnet** | **http** | **username** *user*]

  This command shows the current sessions in use.

  The following is sample output from the **show quota management-session** command:

```
ciscoasa(config)#show quota management-session

#Sessions               ConnectionType                  Username
1                              SSH                      cisco
2                              TELNET                   cisco
1                              SSH                      cisco1
```

- **show aaa login-history** [**user** *name*]

  This command shows the login history per user.

  The following is sample output from the **show aaa login-history** command.

```
ciscoasa(config)# show aaa login-history
Login history for user:    turjan
Logins in last   1 days:   1
Last successful login:     16:44:32 UTC Jul 23 2018 from console
Failures since last login: 0
Last failed login:         None
Privilege level:                                    14
Privilege level changed from 11 to 14 at:           14:07:30 UTC Aug 21 2018
```

History for Management Access

Table 3: History for Management Access

Feature Name	Platform Releases	Description
SSH X.509 certificate authentication	9.20(4)/9.24(1)	You can now use an X.509v3 certificate to authenticate a user for SSH (RFC 6187). For the Firepower 4100/9300, you need FXOS version 2.16(2.109+) or 2.18+. **Note** The bundled version of ASDM 7.20(4) does not include support for this feature. Download and install ASDM 7.20(4) from Cisco.com for feature support. Be sure to change the image name to asdm.bin if you want to overwrite the bundled version. New/Modified commands: **aaa authorization exec ssh-x509**, **ssh authentication method**, **ssh trustpoint sign**, **ssh username-from-certificate**, **validation-usage ssh-client**
ASDM certificate authentication	9.24(1)	ASDM Launcher 1.9(10), which comes with ASDM 7.24, now supports user certificate authentication. Previously, this feature was only supported with Java Web Start (discontinued in 7.18). Because the ASA commands were not deprecated in 9.18, you can configure earlier ASA versions to use certificate authentication when using any ASDM version with ASDM Launcher 1.9(10). New/Modified commands: **http authentication-certificate**, **http username-from-certificate** New/Modified screens: • ASDM Launcher login window.
AES-256-GCM SSH cipher	9.20(4)/9.24(1)	The ASA supports the AES-256-GCM cipher for SSH. It is enabled by default for **all** and **high** encryption levels. New/Modified commands: **ssh cipher encryption**
Message-of-the-day (motd) banner shows the failover state and the last failover time	/9.24(1)	When using failover, if you configure the **banner motd** command, then the banner shows information about the failover state and the last failover time of the unit you are logging into. This information is useful if you are performing actions at the CLI, such as troubleshooting, and a failover occurs between sessions. New or modified commands: **banner motd** *Also in 9.24(1).*

Feature Name	Platform Releases	Description
The ASA SSH stack was deprecated	9.23(1)	You can no longer use the ASA SSH stack. The Cisco SSH stack is now the only stack. New/Modified commands: **ssh stack ciscossh**
CiscoSSH stack now default	9.19(1)	The Cisco SSH stack is now used by default. New/Modified commands: **ssh stack ciscossh**
Loopback interface support for SSH and Telnet	9.18(2)	You can now add a loopback interface and use it for the following features: • SSH • Telnet New/Modified commands: **interface loopback**, **ssh**, **telnet**
CiscoSSH stack	9.17(1)	The ASA uses a proprietary SSH stack for SSH connections. You can now choose to use the CiscoSSH stack instead, which is based on OpenSSH. The default stack continues to be the ASA stack. Cisco SSH supports: • FIPS compliance • Regular updates, including updates from Cisco and the open source community Note that the CiscoSSH stack does not support: • SSH to a different interface over VPN (management-access) • EdDSA key pair • RSA key pair in FIPS mode If you need these features, you should continue to use the ASA SSH stack. There is a small change to SCP functionality with the CiscoSSH stack: to use the ASA **copy** command to copy a file to or from an SCP server, you have to enable SSH access on the ASA for the SCP server subnet/host using the **ssh** command. New/Modified commands: **ssh stack ciscossh**

Feature Name	Platform Releases	Description
Local user lockout changes	9.17(1)	The ASA can lock out local users after a configurable number of failed login attempts. This feature did not apply to users with privilege level 15. Also, a user would be locked out indefinitely until an admin unlocked their account. Now, users will be unlocked after 10 minutes unless an admin uses the **clear aaa local user lockout** command before then. Privilege level 15 users are also now affected by the lockout setting. New/Modified commands: **aaa local authentication attempts max-fail** , **show aaa local user**
SSH and Telnet password change prompt	9.17(1)	The first time a local user logs into the ASA using SSH or Telnet, they are prompted to change their password. They will also be prompted for the first login after an admin changes their password. If the ASA reloads, however, users will not be prompted even if it is their first login. Note that any service that uses the local user database, such as VPN, will also have to use the new password if it was changed during an SSH or Telnet login. New/Modified commands: **show aaa local user**
SSH security improvements	9.16(1)	SSH now supports the following security improvements: • Host key format—**crypto key generate** {**eddsa** | **ecdsa**}. In addition to RSA, we added support for the EdDSA and ECDSA host keys. The ASA tries to use keys in the following order if they exist: EdDSA, ECDSA, and then RSA. If you explicitly configure the ASA to use the RSA key with the **ssh key-exchange hostkey rsa** command, you must generate a key that is 2048 bits or higher. For upgrade compatibility, the ASA will use smaller RSA host keys only when the default host key setting is used. RSA support will be removed in a later release. • Key exchange algorithms—**ssh key-exchange group** {**ecdh-sha2-nistp256** | **curve25519-sha256**} • Encryption algorithms—**ssh cipher encryption chacha20-poly1305@openssh.com** • SSH version 1 is no longer supported—The **ssh version** command is removed. New/Modified commands: **crypto key generate eddsa**, **crypto key zeroize eddsa**, **show crypto key mypubkey**, **ssh cipher encryption chacha20-poly1305@openssh.com**, **ssh key-exchange group** {**ecdh-sha2-nistp256** | **curve25519-sha256**}, **ssh key-exchange hostkey**, **ssh version**
Management access for SNMP	9.14(2)	When configuring management access over a VPN tunnel, include the IP address of the outside interface in the crypto map access-list as part of the VPN configuration for secure SNMP polling over a site-to-site VPN.

Feature Name	Platform Releases	Description
HTTPS idle timeout setting	9.14(1)	You can now set the idle timeout for all HTTPS connections to the ASA, including ASDM, WebVPN, and other clients. Formerly, using the **http server idle-timeout** command, you could only set the ASDM idle timeout. If you set both timeouts, the new command takes precendence. New/Modified commands: **http connection idle-timeout**
SSH encryption ciphers are now listed in order from highest to lowest security for pre-defined lists	9.13(1)	SSH encryption ciphers are now listed in order from highest security to lowest security for pre-defined lists (such as medium or high). In earlier releases, they were listed from lowest to highest, which meant that a low security cipher would be proposed before a high security cipher. New/Modified commands: **ssh cipher encryption**
Setting the SSH key exchange mode is restricted to the Admin context	9.12(2)	You must set the SSH key exchange in the Admin context; this setting is inherited by all other contexts. New/Modified commands: **ssh key-exchange**
enable password change now required on login	9.12(1)	The default **enable** password is blank. When you try to access privileged EXEC mode on the ASA, you are now required to change the password to a value of 3 characters or longer. You cannot keep it blank. The **no enable password** command is no longer supported. At the CLI, you can access privileged EXEC mode using the **enable** command, the **login** command (with a user at privilege level 2+), or an SSH or Telnet session when you enable **aaa authorization exec auto-enable**. All of these methods require you to set the enable password. This password change requirement is not enforced for ASDM logins. In ASDM, by default you can log in without a username and with the **enable** password. New/Modified commands: **enable password**
Configurable limitation of admin sessions	9.12(1)	You can configure the maximum number of aggregate, per user, and per-protocol administrative sessions. Formerly, you could configure only the aggregate number of sessions. This feature does not affect console sessions. Note that in multiple context mode, you cannot configure the number of HTTPS sessions, where the maximum is fixed at 5 sessions. The **quota management-session** command is also no longer accepted in the system configuration, and is instead available in the context configuration. The maximum aggregate sessions is now 15; if you configured 0 (unlimited) or 16+, then when you upgrade, the value is changed to 15. New/Modified commands: **quota management-session**, **show quota management-session**

Feature Name	Platform Releases	Description
Notifications for administrative privilege level changes	9.12(1)	When you authenticate for enable access (**aaa authentication enable console**) or allow privileged EXEC access directly (**aaa authorization exec auto-enable**), then the ASA now notifies users if their assigned access level has changed since their last login. New/Modified commands: **show aaa login-history**
SSH stronger security	9.12(1)	See the following SSH security improvements: • Diffie-Hellman Group 14 SHA256 key exchange support. This setting is now the default. The former default was Group 1 SHA1. • HMAC-SHA256 integrity cipher support. The default is now the high security set of ciphers (hmac-sha2-256 only). The former default was the medium set. New/Modified commands: **ssh cipher integrity** , **ssh key-exchange group dh-group14-sha256**
Allow non-browser-based HTTPS clients to access the ASA	9.12(1)	You can allow non-browser-based HTTPS clients to access HTTPS services on the ASA. By default, ASDM, CSM, and REST API are allowed. New/Modified commands: **http server basic-auth-client**
RSA key pair supports 3072-bit keys	9.9(2)	You can now set the modulus size to 3072. New or modified command: **crypto key generate rsa modulus**
VPN management access on Bridged Virtual Interfaces (BVIs)	9.9(2)	You can now enable management services, such as **telnet**, **http**, and **ssh**, on a BVI if VPN **management-access** has been enabled on that BVI. For non-VPN management access, you should continue to configure these services on the bridge group member interfaces. New or Modified commands: **https**, **telnet**, **ssh**, **management-access**
SSH version 1 has been deprecated	9.9(1)	SSH version 1 has been deprecated, and will be removed in a future release. The default setting has changed from both SSH v1 and v2 to just SSH v2. New/Modified commands: **ssh version**

Feature Name	Platform Releases	Description
Separate authentication for users with SSH public key authentication and users with passwords	9.6(3)/9.8(1)	In releases prior to 9.6(2), you could enable SSH public key authentication (**ssh authentication**) without also explicitly enabling AAA SSH authentication with the Local user database (**aaa authentication ssh console LOCAL**). In 9.6(2), the ASA required you to explicitly enable AAA SSH authentication. In this release, you no longer have to explicitly enable AAA SSH authentication; when you configure the **ssh authentication** command for a user, local authentication is enabled by default for users with this type of authentication. Moreover, when you explicitly configure AAA SSH authentication, this configuration only applies for for usernames with *passwords*, and you can use any AAA server type (**aaa authentication ssh console radius_1**, for example). For example, some users can use public key authentication using the local database, and other users can use passwords with RADIUS. We did not modify any commands.
Login history	9.8(1)	By default, the login history is saved for 90 days. You can disable this feature or change the duration, up to 365 days. This feature only applies to usernames in the local database when you enable local AAA authentication for one or more of the management methods (SSH, ASDM, Telnet, and so on). We introduced the following commands: **aaa authentication login-history, show aaa login-history**
Password policy enforcement to prohibit the reuse of passwords, and prohibit use of a password matching a username	9.8(1)	You can now prohibit the reuse of previous passwords for up to 7 generations, and you can also prohibit the use of a password that matches a username. We introduced the following commands: **password-history, password-policy reuse-interval, password-policy username-check**
ASA SSL Server mode matching for ASDM	9.6(2)	For an ASDM user who authenticates with a certificate, you can now require the certificate to match a certificate map. We modified the following command: **http authentication-certificate match**
SSH public key authentication improvements	9.6(2)	In earlier releases, you could enable SSH public key authentication (**ssh authentication**) without also enabling AAA SSH authentication with the Local user database (**aaa authentication ssh console LOCAL**). The configuration is now fixed so that you must explicitly enable AAA SSH authentication. To disallow users from using a password instead of the private key, you can now create a username without any password defined. We modified the following commands: **ssh authentication, username**
ASDM management authorization	9.4(1)	You can now configure management authorization separately for HTTP access vs. Telnet and SSH access. We introduced the following command: **aaa authorization http console**

Feature Name	Platform Releases	Description
ASDM username from certificate configuration	9.4(1)	When you enable ASDM certificate authentication (**http authentication-certificate**), you can configure how ASDM extracts the username from the certificate; you can also enable pre-filling the username at the login prompt. We introduced the following command: **http username-from-certificate**
Improved one-time password authentication	9.2(1)	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**.
HTTP redirect support for IPV6	9.1(7)/9.6(1)	When you enable HTTP redirect to HTTPS for ASDM access or clientless SSL VPN, you can now redirect traffic sent an to IPv6 address. We added functionality to the following command: **http redirect**
Configurable SSH encryption and integrity ciphers	9.1(7.4)/9.3(3)/9.6(1)	Users can select cipher modes when doing SSH encryption management and can configure HMAC and encryption for varying key exchange algorithms. You might want to change the ciphers to be more or less strict, depending on your application. Note that the performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use **ssh cipher encryption custom aes128-cbc**, for example. We introduced the following commands: **ssh cipher encryption, ssh cipher integrity**.
AES-CTR encryption for SSH	9.1(2)	The SSH server implementation in the ASA now supports AES-CTR mode encryption.
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**.
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.

Feature Name	Platform Releases	Description
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**.
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.

Feature Name	Platform Releases	Description
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**.