Local Database for AAA

This chapter describes how to configure local servers for AAAand includes the following sections:

Information About the Local Database

You can use the local database for the following functions:

- ASDM per-user access
- Console authentication
- Telnet and SSH authentication
- **enable** command authentication

  This setting is for CLI-access only and does not affect the ASDM login.
- Command authorization

  If you turn on command authorization using the local database, then the ASA refers to the user privilege level to determine which commands are available. Otherwise, the privilege level is not generally used. By default, all commands are either privilege level 0 or level 15.
- Network access authentication
- VPN client authentication

For multiple context mode, you can configure usernames in the system execution space to provide individual logins at the CLI using the **login** command; however, you cannot configure any AAA rules that use the local database in the system execution space.

Note You cannot use the local database for network access authorization.

Fallback Support

The local database can act as a fallback method for several functions. This behavior is designed to help you prevent accidental lockout from the ASA.

When a user logs in, the servers in the group are accessed one at a time, starting with the first server that you specify in the configuration, until a server responds. If all servers in the group are unavailable, the ASA tries the local database if you have configured it as a fallback method (for management authentication and authorization only). If you do not have a fallback method, the ASA continues to try the AAA servers.

For users who need fallback support, we recommend that their usernames and passwords in the local database match their usernames and passwords on the AAA servers. This practice provides transparent fallback support. Because the user cannot determine whether a AAA server or the local database is providing the service, using usernames and passwords on AAA servers that are different than the usernames and passwords in the local database means that the user cannot be certain which username and password should be given.

The local database supports the following fallback functions:

- Console and enable password authentication—If the servers in the group are all unavailable, the ASA uses the local database to authenticate administrative access, which can also include enable password authentication.
- Command authorization—If the TACACS+ servers in the group are all unavailable, the local database is used to authorize commands based on privilege levels.
- VPN authentication and authorization—VPN authentication and authorization are supported to enable remote access to the ASA if AAA servers that normally support these VPN services are unavailable. When a VPN client of an administrator specifies a tunnel group configured to fallback to the local database, the VPN tunnel can be established even if the AAA server group is unavailable, provided that the local database is configured with the necessary attributes.

How Fallback Works with Multiple Servers in a Group

If you configure multiple servers in a server group and you enable fallback to the local database for the server group, fallback occurs when no server in the group responds to the authentication request from the ASA. To illustrate, consider this scenario:

You configure an LDAP server group with two Active Directory servers, server 1 and server 2, in that order. When the remote user logs in, the ASA attempts to authenticate to server 1.

If server 1 responds with an authentication failure (such as *user not found*), the ASA does not attempt to authenticate to server 2.

If server 1 does not respond within the timeout period (or the number of authentication attempts exceeds the configured maximum), the ASA tries server 2.

If both servers in the group do not respond, and the ASA is configured to fall back to the local database, the ASA tries to authenticate to the local database.

Licensing Requirements for the Local Database

Model	License Requirement
ASAv	Standard or Premium License.
All other models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

To prevent lockout from the ASA when using the local database for authentication or authorization, see Recovering from a Lockout, page 43-36.

Adding a User Account to the Local Database

To add a user to the local database, perform the following steps:

Detailed Steps

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td><b>Step 1</b></td><td><b>username</b> <i>username</i> {<b>nopassword</b> | <b>password</b> <i>password</i>} [<b>privilege</b> <i>priv_level</i>]

<b>Example:</b>
ciscoasa(config)# username exampleuser1 privilege 1</td><td>Creates the user account. The <b>username</b> <i>username</i> keyword is a string from 4 to 64 characters long.

The <b>password</b> <i>password</i> keyword is a string from 3 to 32 characters long. The <b>privilege</b> <i>level</i> argument sets the privilege level, which ranges from 0 to 15. The default is 2. This privilege level is used with command authorization.

<b>Caution</b> If you do not use command authorization (the <b>aaa authorization console LOCAL</b> command), then the default level 2 allows management access to privileged EXEC mode. If you want to limit access to privileged EXEC mode, either set the privilege level to 0 or 1, or use the <b>service-type</b> command.

The <b>nopassword</b> keyword creates a user account with no password.

The <b>encrypted</b> keyword indicates that the password is encrypted. When you define a password in the <b>username</b> command, the ASA encrypts it when it saves it to the configuration for security purposes. When you enter the <b>show running-config</b> command, the <b>username</b> command does not show the actual password; it shows the encrypted password followed by the <b>encrypted</b> keyword. For example, if you enter the password “test,” the <b>show running-config</b> output would appear as something similar to the following:

<code>username user1 password DLaUiAX3l78qgoB5c7iVNw== encrypted</code>

The only time you would actually enter the <b>encrypted</b> keyword at the CLI is if you are cutting and pasting a configuration file for use in another ASA, and you are using the same password.</td></tr>
</table>

	Command	Purpose
Step 2	**username** *username* **attributes** **Example:** `ciscoasa(config)# username exampleuser1 attributes`	(Optional) Configures username attributes. The *username* argument is the username that you created in Step 1. By default, VPN users that you add with this command have no attributes or group policy association. You must configure all values explicitly using the **username attributes** command. For more information, see the VPN configuration guide.

Step 3	**service-type** {**admin** | **nas-prompt** | **remote-access**}
	Example:
	`ciscoasa(config-username)# service-type admin`
		(Optional) Configures the user level if you configured management authorization using the **aaa authorization exec** command (see Limiting User CLI and ASDM Access with Management Authorization, page 43-23). The **admin** keyword allows full access to any services specified by the **aaa authentication console LOCAL** commands. The **admin** keyword is the default.
		The **nas-prompt** keyword allows access to the CLI when you configure the **aaa authentication** {**telnet | ssh | serial**} **console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you enable authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command (or the **login** command).
		The **remote-access** keyword denies management access. The user cannot use any services specified by the **aaa authentication console** commands (excluding the **serial** keyword; serial access is allowed).

	Command	Purpose

Step 4	**`ssh authentication`** `{`**`pkf`** `|` **`publickey`** *`key`* `[`**`hashed`**`]}`
	
	Example:
	`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"`
	`AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza371B/Q/fljp`
	`LAv1BbyAd5PJCJXh/U4LO`
	`hleR/qgIROjpnFaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPh`
	`PHCi0hIt4oUF2ZbXESA/8`
	`jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCS`
	`Tx9QC//wt6E/zRcdoqiJG`
	`p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9q`
	`D3MqsV+PkJGSGiqZwnyIl`
	`QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTf`
	`fIh6O+xKh93gwTgzaZTK4`
	`CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw`
	`9WUg/rapekKloz3tsPTDe`
	`p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMP`
	`YJl+xgKAkuHDkBlMS4i8b`
	`Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzr`
	`QT2mXBcSKQNWlSCBpCHsk`
	`/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq`
	`0Rjo34+61+70PCtYXebxM`
	`Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVV`
	`M1Qqw1uL4r99CbZF9NghY`
	`NRxCQOY/7K77II==`
	`---- END SSH2 PUBLIC KEY ----quit`
	`INFO: Import of an SSH public key formatted file`
	`SUCCEEDED.`
	`ciscoasa(config-username)#`
		Enables public key authentication for SSH connections to the ASA on a per-user basis. You can specify a public key file (PKF) formatted key (the **pkf** keyword) or a Base64 key (the **publickey** keyword).
		
		For a **publickey**, 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 raw keys (with no certificates).
		
		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.
		
		When you view the key on the ASA using the **show running-config username** command, the key is encrypted using a SHA-256 hash. Even if you entered the key as **pkf**, the ASA hashes the key, and shows it as a hashed **publickey**. If you need to copy the key from **show** output, specify the **publickey** type with the **hashed** keyword.

Step 5	(Optional) If you are using this username for VPN authentication, you can configure many VPN attributes for the user. For more information, see the VPN configuration guide.	

Examples

The following example assigns a privilege level of 15 to the admin user account:

```
ciscoasa(config)# username admin password password privilege 15
```

The following example creates a user account with no password:

```
ciscoasa(config)# username user34 nopassword
```

The following example enables management authorization, creates a user account with a password, enters username configuration mode, and specifies a **service-type** of **nas-prompt**:

```
ciscoasa(config)# aaa authorization exec authentication-server
ciscoasa(config)# username user1 password gOgeOus
ciscoasa(config)# username user1 attributes
ciscoasa(config-username)# service-type nas-prompt
```

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

Step 1 Generate the ssh-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           |
+-----------------+
```

Step 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 ramona@rboersma-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$
```

Step 3 Copy the key to your clipboard.

Step 4 Connect to the ASA CLI, and add the public key to your username:

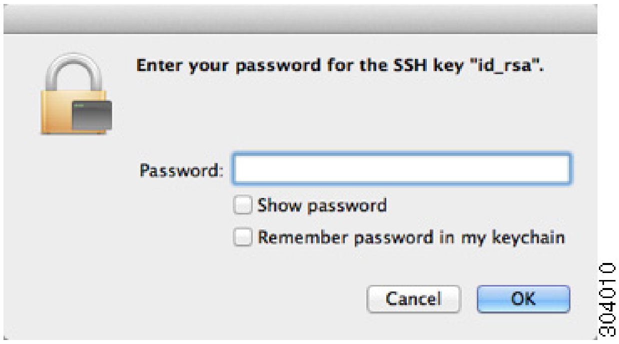
```
ciscoasa(config)# username test attributes
ciscoasa(config-username)# ssh authentication pkf
Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by ramona@rboersma-mac from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
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 ----
quit
INFO: Import of an SSH public key formatted file completed successfully.
```

Step 5 Verify the user (test) can SSH to the ASA:

```
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:


Enter your password for the SSH key "id_rsa".

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>
```

Monitoring the Local Database

To monitor the local database,enter one of the following commands:

Command	Purpose
show aaa-server	Shows the configured database statistics. To clear the AAA server configuration, enter the **clear aaa-server statistics** command.
show running-config aaa-server	Shows the AAA server running configuration. To clear AAA server statistics, enter the **clear configure aaa-server** command.

Feature History for the Local Database

Table 35-1 lists each feature change and the platform release in which it was implemented.

Table 35-1 Feature History for the Local Database

Feature Name	Platform Releases	Feature Information
Local database configuration for AAA	7.0(1)	Describes how to configure the local database for AAA use. We introduced the following commands: **username**, **aaa authorization exec authentication-server**, **aaa authentication console LOCAL**, **aaa authorization exec LOCAL**, **service-type, aaa authentication** {**telnet** | **ssh** | **serial**} **console LOCAL**, **aaa authentication http console LOCAL**, **aaa authentication enable console LOCAL**, **show running-config aaa-server**, **show aaa-server**, **clear configure aaa-server**, **clear aaa-server statistics**.
Support for SSH public key authentication	9.1(2)	You can now 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**. *Also available in 8.4(4.1); PKF key format support is only in 9.1(2).*