

AAA and the Local Database

This chapter describes authentication, authorization, and accounting (AAA, pronounced “triple A”). AAA is a a set of services for controlling access to computer resources, enforcing policies, assessing usage, and providing the information necessary to bill for services. These processes are considered important for effective network management and security.

This chapter also describes how to configure the local database for AAA functionality. For external AAA servers, see the chapter for your server type.

- About AAA and the Local Database, on page 1
- Guidelines for the Local Database, on page 6
- Add a User Account to the Local Database, on page 6
- Monitoring the Local Database, on page 8
- History for the Local Database, on page 9

About AAA and the Local Database

This section describes AAA and the local database.

Authentication

Authentication provides a way to identify a user, typically by having the user enter a valid username and valid password before access is granted. The AAA server compares a user's authentication credentials with other user credentials stored in a database. If the credentials match, the user is permitted access to the network. If the credentials do not match, authentication fails and network access is denied.

You can configure the ASA to authenticate the following items:

- All administrative connections to the ASA, including the following sessions:
  - Telnet
  - SSH
  - Serial console
  - ASDM using HTTPS
  - VPN management access

- The **enable** command
- Network access
- VPN access

Authorization

Authorization is the process of enforcing policies: determining what types of activities, resources, or services a user is permitted to access. After a user is authenticated, that user may be authorized for different types of access or activity.

You can configure the ASA to authorize the following items:

- Management commands
- Network access
- VPN access

Accounting

Accounting measures the resources a user consumes during access, which may include the amount of system time or the amount of data that a user has sent or received during a session. Accounting is carried out through the logging of session statistics and usage information, which is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.

Interaction Between Authentication, Authorization, and Accounting

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

AAA Servers and Server Groups

The AAA server is a network server that is used for access control. Authentication identifies the user. Authorization implements policies that determine which resources and services an authenticated user may access. Accounting keeps track of time and data resources that are used for billing and analysis.

If you want to use an external AAA server, you must first create a AAA server group for the protocol that the external server uses, and add the server to the group. You can create more than one group per protocol, and separate groups for all protocols that you want to use. Each server group is specific to one type of server or service.

See the following topics for details on how to create the groups:

- Configure RADIUS Server Groups
- Configure TACACS+ Server Groups
- Configure LDAP Server Groups
- Configure Kerberos AAA Server Groups

- Configure RSA SecurID AAA Server Groups

See the VPN configuration guide for more information on using Kerberos Constrained Delegation and HTTP Form.

The following table summarizes the supported types of server and their uses, including the local database.

Table 1: Supported Services for AAA Servers

Server Type and Service	Authentication	Authorization	Accounting
Local Database			
Administrators	Yes	Yes	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	Yes	Yes	No
RADIUS			
Administrators	Yes	Yes	Yes
VPN Users	Yes	Yes	Yes
Firewall Sessions (AAA rules)	Yes	Yes	Yes
TACACS+			
Administrators	Yes	Yes	Yes
VPN Users	Yes	No	Yes
Firewall Sessions (AAA rules)	Yes	Yes	Yes
LDAP			
Administrators	Yes	No	No
VPN Users	Yes	Yes	No
Firewall Sessions (AAA rules)	Yes	No	No
Kerberos			
Administrators	Yes	No	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	Yes	No	No
SDI (RSA SecurID)			
Administrators	Yes	No	No
VPN Users	Yes	No	No

Server Type and Service	Authentication	Authorization	Accounting
Firewall Sessions (AAA rules)	Yes	No	No
HTTP Form			
Administrators	No	No	No
VPN Users	Yes	No	No
Firewall Sessions (AAA rules)	No	No	No

Notes

- RADIUS—Accounting for administrators does not include command accounting.
- RADIUS—Authorization for firewall sessions is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response.
- TACACS+—Accounting for administrators includes command accounting.
- HTTP Form—Authentication and SSO operations for clientless SSL VPN user sessions only.

About the Local Database

The ASA maintains a local database that you can populate with user profiles. You can use a local database instead of AAA servers to provide user authentication, authorization, and accounting.

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 Cisco 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.

Note A RADIUS client, by default, sends three requests before timing out. When the client attempts to SSH, the request is sent to one server at a time. As the requests time out, the server is marked as FAILED. The next RADIUS request goes to the next ACTIVE server, subsequently marking all the servers as FAILED, and eventually falling back to LOCAL authentication (if configured). The servers marked as FAILED eventually recover and are made ACTIVE. In this case, if you have 3-4 servers in the group, you might observe intermittent login issues.

Guidelines for the Local Database

Make sure that you prevent a lockout from the ASA when using the local database for authentication or authorization.

Add a User Account to the Local Database

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

Procedure

Step 1 Create the user account.

username *username* [**password** *password*] [**privilege** *priv_level*]

Example:

```
ciscoasa(config)# username exampleuser1 password madmaxfuryroadrules privilege 1
```

The**username** *username* keyword is a string from 3 to 64 characters long, using any combination of ASCII printable characters (character codes 32-126), with the exception of spaces and the question mark. The **password** *password* keyword is a string from 3 to 127 characters long, and can be any combination of ASCII printable characters (character codes 32-126), with the following exceptions:

- No spaces
- No question marks

You might want to create a username without a password if you are using SSH public key authentication, for example. The **privilege** *priv_level* keyword sets the privilege level, which ranges from 0 to 15. The default is 2. This privilege level is used with command authorization.

Caution
If you do not use command authorization (the **aaa authorization console LOCAL** 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 **service-type** command.

These less-used options are not shown in the above syntax: The **nopassword** keyword creates a user account without a password; this option is insecure, is not recommended, and is not supported with all types of connections.

The **encrypted** keyword (for passwords 32 characters and fewer in 9.6 and earlier) or the **pbkdf2** keyword (for passwords longer than 32 characters in 9.6 and later, and passwords of all lengths in 9.7 and later) indicates that the password is encrypted (using an MD5-based hash or a PBKDF2 (Password-Based Key Derivation Function 2) hash). Note that already existing passwords continue to use the MD5-based hash unless you enter a new password. When you define a password in the **username** command, the ASA encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **username** command does not show the actual password; it shows the encrypted password followed by the **encrypted** or **pbkdf2** keyword. For example, if you enter the password "test," the **show running-config** command output would appear as something similar to the following:

```
username user1 password DLaUiAX3l78qgoB5c7iVNw== encrypted
```

The only time you would actually enter the **encrypted** or **pbkdf2** 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.

Step 2 (Optional) Configure username attributes.

username *username* **attributes**

Example:

```
ciscoasa(config)# username exampleuser1 attributes
```

The *username* argument is the username that you created in the first step.

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. See the VPN configuration guide for more information.

Step 3 (Optional) Configure the user level if you configured management authorization using the **aaa authorization exec** command.

service-type {**admin** | **nas-prompt** | **remote-access**}

Example:

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

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. You cannot use any services specified by the **aaa authentication console** commands (excluding the **serial** keyword; serial access is allowed).

Step 4 (Optional) For public key authentication for SSH connections to the ASA on a per-user basis, see Configure SSH for Public Key Access.

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

Examples

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

```
ciscoasa(config)# username admin password farscape1 privilege 15
```

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 gOrgeOus
ciscoasa(config)# username user1 attributes
ciscoasa(config-username)# service-type nas-prompt
```

Monitoring the Local Database

See the following commands for monitoring the local database:

- **show aaa-server**

  This command shows the configured database statistics. To clear the AAA server statistics, enter the **clear aaa-server statistics** command.

- **show running-config aaa-server**

  This command shows the AAA server running configuration. To clear AAA server configuration, enter the **clear configure aaa-server** command.

History for the Local Database

Table 2: History for the Local Database

Feature Name	Platform Releases	Description
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).*
Longer password support for local **username** and **enable** passwords (up to 127 characters)	9.6(1)	You can now create local **username** and **enable** passwords up to 127 characters (the former limit was 32). When you create a password longer than 32 characters, it is stored in the configuration using a PBKDF2 (Password-Based Key Derivation Function 2) hash. Shorter passwords continue to use the MD5-based hashing method. We modified the following commands: **enable, username**
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**

Feature Name	Platform Releases	Description
PBKDF2 hashing for all local **username** and **enable** passwords	9.7(1)	Local **username** and **enable** passwords of all lengths are stored in the configuration using a PBKDF2 (Password-Based Key Derivation Function 2) hash. Previously, passwords 32 characters and shorter used the MD5-based hashing method. Already existing passwords continue to use the MD5-based hash unless you enter a new password. See the "Software and Configurations" chapter in the General Operations Configuration Guide for downgrading guidelines. We modified the following commands: **enable, username**
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.