CHAPTER 35

Configuring AAA Servers and the Local Database

This chapter describes support for authentication, authorization, and accounting (AAA, pronounced "triple A"), and how to configure AAA servers and the local database.

The chapter includes the following sections:

Information About AAA

AAA enables the ASA to determine who the user is (authentication), what the user can do (authorization), and what the user did (accounting).

AAA provides an extra level of protection and control for user access than using access lists alone. For example, you can create an access list allowing all outside users to access Telnet on a server on the DMZ network. If you want only some users to access the server and you might not always know IP addresses of these users, you can enable AAA to allow only authenticated and/or authorized users to connect through the ASA. (The Telnet server enforces authentication, too; the ASA prevents unauthorized users from attempting to access the server.)

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.

This section includes the following topics:

Information About Authentication

Authentication controls access by requiring valid user credentials, which are usually a username and password. 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

Information About Authorization

Authorization controls access *per user* after users are authenticated. You can configure the ASA to authorize the following items:

- Management commands
- Network access
- VPN access

Authorization controls the services and commands that are available to each authenticated user. If you did not enable authorization, authentication alone would provide the same access to services for all authenticated users.

If you need the control that authorization provides, you can configure a broad authentication rule, and then have a detailed authorization configuration. For example, you can authenticate inside users who try to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

The ASA caches the first 16 authorization requests per user, so if the user accesses the same services during the current authentication session, the ASA does not resend the request to the authorization server.

Information About Accounting

Accounting tracks traffic that passes through the ASA, enabling you to have a record of user activity. If you enable authentication for that traffic, you can account for traffic per user. If you do not authenticate the traffic, you can account for traffic per IP address. Accounting information includes session start and stop times, username, the number of bytes that pass through the ASA for the session, the service used, and the duration of each session.

Summary of Server Support

Table 35-1 summarizes the support for each AAA service by each AAA server type, including the local database. For more information about support for a specific AAA server type, see the topics following the table.

Table 35-1 Summary of AAA Support

	Database Type							
AAA Service	**Local**	**RADIUS**	**TACACS+**	**SDI (RSA)**	**NT**	**Kerberos**	**LDAP**	**HTTP Form**
Authentication of...								
VPN users[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[2]
Firewall sessions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Administrators	Yes	Yes	Yes	Yes[3]	Yes	Yes	Yes	No
Authorization of...								
VPN users	Yes	Yes	No	No	No	No	Yes	No
Firewall sessions	No	Yes[4]	Yes	No	No	No	No	No
Administrators	Yes[5]	No	Yes	No	No	No	No	No
Accounting of...								
VPN connections	No	Yes	Yes	No	No	No	No	No
Firewall sessions	No	Yes	Yes	No	No	No	No	No
Administrators	No	Yes[6]	Yes	No	No	No	No	No

1. For SSL VPN connections, either PAP or MS-CHAPv2 can be used.
2. HTTP Form protocol supports both authentication and single sign-on operations for clientless SSL VPN users sessions only.
3. RSA/SDI is supported for ASDM HTTP administrative access with ASA 5500 software version 8.2(1) or later.
4. For firewall sessions, RADIUS authorization is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response.
5. Local command authorization is supported by privilege level only.
6. Command accounting is available for TACACS+ only.

Note In addition to the native protocol authentication listed in Table 35-1, the ASA supports proxying authentication. For example, the ASA can proxy to an RSA/SDI and/or LDAP server via a RADIUS server. Authentication via digital certificates and/or digital certificates with the AAA combinations listed in the table are also supported.

RADIUS Server Support

The ASA supports the following RFC-compliant RADIUS servers for AAA:

- Cisco Secure ACS 3.2, 4.0, 4.1, 4.2, and 5.x
- Cisco Identity Services Engine (ISE)
- RSA RADIUS in RSA Authentication Manager 5.2, 6.1, and 7.x
- Microsoft

Authentication Methods

The ASA supports the following authentication methods with RADIUS:

- PAP—For all connection types.
- CHAP and MS-CHAPv1—For L2TP-over-IPsec connections.
- MS-CHAPv2—For L2TP-over-IPsec connections, and for regular IPsec remote access connections when the password management feature is enabled. You can also use MS-CHAPv2 with clientless connections.
- Authentication Proxy modes—Including RADIUS to Active Directory, RADIUS to RSA/SDI, RADIUS to Token-server, and RSA/SDI to RADIUS connections,

Note To enable MS-CHAPv2 as the protocol used between the ASA and the RADIUS server for a VPN connection, password management must be enabled in the tunnel group general attributes. Enabling password management generates an MS-CHAPv2 authentication request from the ASA to the RADIUS server. See the description of the **password-management** command for details.

If you use double authentication and enable password management in the tunnel group, then the primary and secondary authentication requests include MS-CHAPv2 request attributes. If a RADIUS server does not support MS-CHAPv2, then you can configure that server to send a non-MS-CHAPv2 authentication request by using the **no mschapv2-capable** command.

Attribute Support

The ASA supports the following sets of RADIUS attributes:

- Authentication attributes defined in RFC 2138.
- Accounting attributes defined in RFC 2139.
- RADIUS attributes for tunneled protocol support, defined in RFC 2868.
- Cisco IOS Vendor-Specific Attributes (VSAs), identified by RADIUS vendor ID 9.
- Cisco VPN-related VSAs, identified by RADIUS vendor ID 3076.
- Microsoft VSAs, defined in RFC 2548.
- Cisco VSA (Cisco-Priv-Level), which provides a standard 0-15 numeric ranking of privileges, with 1 being the lowest level and 15 being the highest level. A zero level indicates no privileges. The first level (login) allows privileged EXEC access for the commands available at this level. The second level (enable) allows CLI configuration privileges.

- A list of attributes is available at the following URL: http://www.cisco.com/en/US/docs/security/asa/asa84/configuration/guide/ref_extserver.html#wp1605508

RADIUS Authorization Functions

The ASA can use RADIUS servers for user authorization of VPN remote access and firewall cut-through-proxy sessions using dynamic access lists or access list names per user. To implement dynamic access lists, you must configure the RADIUS server to support it. When the user authenticates, the RADIUS server sends a downloadable access list or access list name to the ASA. Access to a given service is either permitted or denied by the access list. The ASA deletes the access list when the authentication session expires.

In addtition to access lists, the ASA supports many other attributes for authorization and setting of permissions for VPN remote access and firewall cut-through proxy sessions. For a complete list of authorization attributes, see the following URL: http://www.cisco.com/en/US/docs/security/asa/asa84/configuration/guide/ref_extserver.html#wp1605508

TACACS+ Server Support

The ASA supports TACACS+ authentication with ASCII, PAP, CHAP, and MS-CHAPv1.

RSA/SDI Server Support

The RSA SecureID servers are also known as SDI servers.

This section includes the following topics:

RSA/SDI Version Support

The ASA supports SDI Versions 5.x, 6.x, and 7.x. SDI uses the concepts of an SDI primary and SDI replica servers. Each primary and its replicas share a single node secret file. The node secret file has its name based on the hexadecimal value of the ACE or Server IP address, with .sdi appended.

A version 5.x, 6.x, or 7.x SDI server that you configure on the ASA can be either the primary or any one of the replicas. See the "RSA/SDI Primary and Replica Servers" section on page 35-6 for information about how the SDI agent selects servers to authenticate users.

Two-step Authentication Process

SDI Versions 5.x, 6.x, or 7.x use a two-step process to prevent an intruder from capturing information from an RSA SecurID authentication request and using it to authenticate to another server. The agent first sends a lock request to the SecurID server before sending the user authentication request. The server

locks the username, preventing another (replica) server from accepting it. This actions means that the same user cannot authenticate to two ASAs using the same authentication servers simultaneously. After a successful username lock, the ASA sends the passcode.

RSA/SDI Primary and Replica Servers

The ASA obtains the server list when the first user authenticates to the configured server, which can be either a primary or a replica. The ASA then assigns priorities to each of the servers on the list, and subsequent server selection is derived at random from those assigned priorities. The highest priority servers have a higher likelihood of being selected.

NT Server Support

The ASA supports Microsoft Windows server operating systems that support NTLM Version 1, collectively referred to as NT servers.

Note NT servers have a maximum length of 14 characters for user passwords. Longer passwords are truncated, which is a limitation of NTLM Version 1.

Kerberos Server Support

The ASA supports 3DES, DES, and RC4 encryption types.

Note The ASA does not support changing user passwords during tunnel negotiation. To avoid this situation happening inadvertently, disable password expiration on the Kerberos/Active Directory server for users connecting to the ASA.

For a simple Kerberos server configuration example, see Example 35-2 on page 35-16.

LDAP Server Support

The ASA supports LDAP. This section includes the following topics:

Authentication with LDAP

During authentication, the ASA acts as a client proxy to the LDAP server for the user, and authenticates to the LDAP server in either plain text or by using the SASL protocol. By default, the ASA passes authentication parameters, usually a username and password, to the LDAP server in plain text.

The ASA supports the following SASL mechanisms, listed in order of increasing strength:

- Digest-MD5—The ASA responds to the LDAP server with an MD5 value computed from the username and password.

- Kerberos—The ASA responds to the LDAP server by sending the username and realm using the GSSAPI Kerberos mechanism.

You can configure the ASA and LDAP server to support any combination of these SASL mechanisms. If you configure multiple mechanisms, the ASA retrieves the list of SASL mechanisms that are configured on the server and sets the authentication mechanism to the strongest mechanism configured on both the ASA and the server. For example, if both the LDAP server and the ASA support both mechanisms, the ASA selects Kerberos, the stronger of the mechanisms.

When user LDAP authentication has succeeded, the LDAP server returns the attributes for the authenticated user. For VPN authentication, these attributes generally include authorization data that is applied to the VPN session. Thus, using LDAP accomplishes authentication and authorization in a single step.

LDAP Server Types

The ASA supports LDAP version 3 and is compatible with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server), the Microsoft Active Directory, Novell, OpenLDAP, and other LDAPv3 directory servers.

By default, the ASA auto-detects whether it is connected to Microsoft Active Directory, Sun LDAP, Novell, OpenLDAP, or a generic LDAPv3 directory server. However, if auto-detection fails to determine the LDAP server type, and you know the server is either a Microsoft, Sun or generic LDAP server, you can manually configure the server type.

When configuring the server type, note the following guidelines:

- The DN configured on the ASA to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACL on the default password policy.
- You must configure LDAP over SSL to enable password management with Microsoft Active Directory and Sun servers.
- The ASA does not support password management with Novell, OpenLDAP, and other LDAPv3 directory servers.
- The ASA uses the Login Distinguished Name (DN) and Login Password to establish a trust relationship (bind) with an LDAP server. For more information, see the "Binding the ASA to the LDAP Server" section on page C-4.

HTTP Forms Authentication for Clientless SSL VPN

The ASA can use the HTTP Form protocol for both authentication and single sign-on (SSO) operations of Clientless SSL VPN user sessions only. For configuration information, see the "Using Single Sign-on with Clientless SSL VPN" section on page 72-21.

Local Database Support, Including as a Falback Method

The ASA maintains a local database that you can populate with user profiles.

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.

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.

Using Certificates and User Login Credentials

The following section describes the different methods of using certificates and user login credentials (username and password) for authentication and authorization. These methods apply to IPsec, AnyConnect, and Clientless SSL VPN.

In all cases, LDAP authorization does not use the password as a credential. RADIUS authorization uses either a common password for all users or the username as a password.

This section includes the following topics:

Using User Login Credentials

The default method for authentication and authorization uses the user login credentials.

- Authentication
  - Enabled by the authentication server group setting in the tunnel group (also called ASDM Connection Profile)
  - Uses the username and password as credentials
- Authorization
  - Enabled by the authorization server group setting in the tunnel group (also called ASDM Connection Profile)
  - Uses the username as a credential

Using Certificates

If user digital certificates are configured, the ASA first validates the certificate. It does not, however, use any of the DNs from certificates as a username for the authentication.

If both authentication and authorization are enabled, the ASA uses the user login credentials for both user authentication and authorization.

- Authentication
  - Enabled by the authentication server group setting
  - Uses the username and password as credentials
- Authorization
  - Enabled by the authorization server group setting
  - Uses the username as a credential

If authentication is disabled and authorization is enabled, the ASA uses the primary DN field for authorization.

- Authentication
  - DISABLED (set to None) by the authentication server group setting
  - No credentials used
- Authorization
  - Enabled by the authorization server group setting

– Uses the username value of the certificate primary DN field as a credential

Note If the primary DN field is not present in the certificate, the ASA uses the secondary DN field value as the username for the authorization request.

For example, consider a user certificate that includes the following Subject DN fields and values:

```
Cn=anyuser,OU=sales;O=XYZCorporation;L=boston;S=mass;C=us;ea=anyuser@example.com
```

If the Primary DN = EA (E-mail Address) and the Secondary DN = CN (Common Name), then the username used in the authorization request would be anyuser@example.com.

Licensing Requirements for AAA Servers

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

The **username** command has two versions: one for 8.4(3) and earlier and one for 8.4(4.1) and later. See the command reference for more information.

Configuring AAA

This section includes the following topics:

Task Flow for Configuring AAA

Step 1 Do one or both of the following:

- Add a AAA server group. See the "Configuring AAA Server Groups" section on page 35-11.
- Add a user to the local database. See the "Adding a User Account to the Local Database" section on page 35-20.

Step 2 (Optional) Configure authorization from an LDAP server that is separate and distinct from the authentication mechanism. See the "Configuring Authorization with LDAP for VPN" section on page 35-16.

Step 3 For an LDAP server, configure LDAP attribute maps. See the "Configuring LDAP Attribute Maps" section on page 35-18.

Step 4 For an administrator, specify the password policy attributes for users. See the "Managing User Passwords" section on page 35-25.

Step 5 (Optional) Users can change their own passwords. See the ".Changing User Passwords" section on page 35-27.

Step 6 (Optional) Users can authenticate with a public key. See the "Authenticating Users with a Public Key for SSH" section on page 35-28.

Step 7 (Optional) Distinguish between administrative and remote-access users when they authenticate. See the "Differentiating User Roles Using AAA" section on page 35-28.

Configuring AAA Server Groups

If you want to use an external AAA server for authentication, authorization, or accounting, you must first create at least one AAA server group per AAA protocol and add one or more servers to each group. You identify AAA server groups by name. Each server group is specific to one type of server: Kerberos, LDAP, NT, RADIUS, SDI, or TACACS+.

Guidelines

- You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode.
- Each group can have up to 16 servers in single mode or 4 servers in multiple mode.
- When a user logs in, the servers are accessed one at a time, starting with the first server 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 configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the ASA continues to try the AAA servers.

Detailed Steps

	Command	Purpose

Step 1	**aaa-server** *server_tag* **protocol** {**kerberos** | **ldap** | **nt** | **radius** | **sdi** | **tacacs+**}
	
	Example:
	`hostname(config)# aaa-server servergroup1 protocol ldap`
	`hostname(config-aaa-server-group)#`
	
	`hostname(config)# aaa-server servergroup1 protocol radius`
	`hostname(config-aaa-server-group)# interim-accounting-update`
	
	`hostname(config)# aaa-server servergroup1 protocol radius`
	`hostname(config-aaa-server-group)# ad-agent-mode`
		Identifies the server group name and the protocol. For example, to use RADIUS to authenticate network access and TACACS+ to authenticate CLI access, you need to create at least two server groups, one for RADIUS servers and one for TACACS+ servers.
		
		You can have up to 100 server groups in single mode or 4 server groups per context in multiple mode. Each group can have up to 15 servers in single mode or 4 servers in multiple mode.
		
		When you enter the **aaa-server protocol** command, you enter aaa-server group configuration mode.
		
		The **interim-accounting-update** option enables multi-session accounting for clientless SSL and AnyConnect sessions. If you choose this option, interim accounting records are sent to the RADIUS server in addition to the start and stop records.
		
		Tip Choose this option if users have trouble completing a VPN connection using clean access SSO, which might occur when making clientless or AnyConnect connections directly to the ASA.
		
		The **ad-agent-mode** option specifies the shared secret between the ASA and the AD agent, and indicates that a RADIUS server group includes AD agents that are not full-function RADIUS servers. Only a RADIUS server group that has been configured using the **ad-agent-mode** option can be associated with user identity. As a result, the **test aaa-server** {**authentication** | **authorization**} *aaa-server-group* command is not available when a RADIUS server group that is not configured using the **ad-agent-mode** option is specified.

	Command	Purpose

Step 2	**merge-dacl** {**before-avpair** | **after-avpair**}
	
	Example:
	`hostname(config)# aaa-server servergroup1 protocol radius`
	`hostname(config-aaa-server-group)# merge-dacl before-avpair`
		Merges a downloadable ACL with the ACL received in the Cisco AV pair from a RADIUS packet. The default setting is **no merge dacl**, which specifies that downloadable ACLs will not be merged with Cisco AV pair ACLs. If both an AV pair and a downloadable ACL are received, the AV pair has priority and is used.
		
		The **before-avpair** option specifies that the downloadable ACL entries should be placed before the Cisco AV pair entries.
		
		The **after-avpair** option specifies that the downloadable ACL entries should be placed after the Cisco AV pair entries. This option applies only to VPN connections. For VPN users, ACLs can be in the form of Cisco AV pair ACLs, downloadable ACLs, and an ACL that is configured on the ASA. This option determines whether or not the downloadable ACL and the AV pair ACL are merged, and does not apply to any ACLs configured on the ASA.

Step 3	**max-failed-attempts** *number* **Example:** `hostname(config-aaa-server-group)# max-failed-attempts 2`	Specifies the maximum number of requests sent to a AAA server in the group before trying the next server. The *number* argument can range from 1 and 5. The default is 3. If you configured a fallback method using the local database (for management access only; see the "Configuring Local Command Authorization" section on page 37-23 and the "Configuring TACACS+ Command Authorization" section on page 37-29 to configure the fallback mechanism), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step. If you do not have a fallback method, the ASA continues to retry the servers in the group.

	Command	Purpose
Step 4	**reactivation-mode** {**depletion** [**deadtime** *minutes*] | **timed**} **Example:** `hostname(config-aaa-server-group)# reactivation-mode deadtime 20`	Specifies the method (reactivation policy) by which failed servers in a group are reactivated. The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive. The **deadtime** *minutes* keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes. The **timed** keyword reactivates failed servers after 30 seconds of down time.
Step 5	**accounting-mode simultaneous** **Example:** `hostname(config-aaa-server-group)# accounting-mode simultaneous`	Sends accounting messages to all servers in the group (RADIUS or TACACS+ only). To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.
Step 6	**aaa-server** *server_group* [*interface_name*] **host** *server_ip* **Example:** `hostname(config)# aaa-server servergroup1 outside host 10.10.1.1`	Identifies the server and the AAA server group to which it belongs. When you enter the **aaa-server host** command, you enter aaa-server host configuration mode. As needed, use host configuration mode commands to further configure the AAA server. The commands in host configuration mode do not apply to all AAA server types. Table 35-2 lists the available commands, the server types to which they apply, and whether or not a new AAA server definition has a default value for that command. Where a command is applicable to the specified server type and no default value is provided (indicated by "—"), use the command to specify the value.

Table 35-2 Host Mode Commands, Server Types, and Defaults

Command	Applicable AAA Server Types	Default Value	Description
accounting-port	RADIUS	1646	
acl-netmask-convert	RADIUS	standard	
authentication-port	RADIUS	1645	
kerberos-realm	Kerberos	—	
key	RADIUS	—	
	TACACS+	—	
ldap-attribute-map	LDAP	—	
ldap-base-dn	LDAP	—	
ldap-login-dn	LDAP	—	

Table 35-2 Host Mode Commands, Server Types, and Defaults (continued)

Command	Applicable AAA Server Types	Default Value	Description
ldap-login-password	LDAP	—	
ldap-naming-attribute	LDAP	—	
ldap-over-ssl	LDAP	636	If not set, the ASA uses sAMAccountName for LDAP requests. Whether using SASL or plain text, you can secure communications between the ASA and the LDAP server with SSL. If you do not configure SASL, we strongly recommend that you secure LDAP communications with SSL.
ldap-scope	LDAP	—	
mschapv2-capable	RADIUS	enabled	
nt-auth-domain-controller	NT	—	
radius-common-pw	RADIUS	—	
retry-interval	Kerberos	10 seconds	
	RADIUS	10 seconds	
	SDI	10 seconds	
sasl-mechanism	LDAP	—	
server-port	Kerberos	88	
	LDAP	389	
	NT	139	
	SDI	5500	
	TACACS+	49	
server-type	LDAP	auto-discovery	If auto-detection fails to determine the LDAP server type, and you know the server is either a Microsoft, Sun or generic LDAP server, you can manually configure the server type.
timeout	All	10 seconds	

Examples

Example 35-1 shows how to add one TACACS+ group with one primary and one backup server, one RADIUS group with a single server, and an NT domain server.

Example 35-1 Multiple AAA Server Groups and Servers

```
hostname(config)# aaa-server AuthInbound protocol tacacs+
hostname(config-aaa-server-group)# max-failed-attempts 2
hostname(config-aaa-server-group)# reactivation-mode depletion deadtime 20
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.2
```

```
hostname(config-aaa-server-host)# key TACPlusUauthKey2
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthOutbound protocol radius
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key RadUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server NTAuth protocol nt
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server NTAuth (inside) host 10.1.1.4
hostname(config-aaa-server-host)# nt-auth-domain-controller primary1
hostname(config-aaa-server-host)# exit
```

Example 35-2 shows how to configure a Kerberos AAA server group named watchdogs, add a AAA server to the group, and define the Kerberos realm for the server. Because Example 35-2 does not define a retry interval or the port that the Kerberos server listens to, the ASA uses the default values for these two server-specific parameters. Table 35-2 lists the default values for all AAA server host mode commands.

Note Kerberos realm names use numbers and upper-case letters only. Although the ASA accepts lower-case letters for a realm name, it does not translate lower-case letters to upper-case letters. Be sure to use upper-case letters only.

Example 35-2 Kerberos Server Group and Server

```
hostname(config)# aaa-server watchdogs protocol kerberos
hostname(config-aaa-server-group)# aaa-server watchdogs host 192.168.3.4
hostname(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
hostname(config-aaa-server-host)# exit
hostname(config)#
```

Configuring Authorization with LDAP for VPN

When user LDAP authentication for VPN access has succeeded, the ASA queries the LDAP server which returns LDAP attributes. These attributes generally include authorization data that applies to the VPN session. Thus, using LDAP accomplishes authentication and authorization in a single step.

There may be cases, however, where you require authorization from an LDAP directory server that is separate and distinct from the authentication mechanism. For example, if you use an SDI or certificate server for authentication, no authorization information is passed back. For user authorizations in this case, you can query an LDAP directory after successful authentication, accomplishing authentication and authorization in two steps.

To set up VPN user authorization using LDAP, perform the following steps.

Detailed Steps

	Command	Purpose

Step 1	**aaa-server** *server_group* **protocol** {**kerberos** | **ldap** | **nt** | **radius** | **sdi** | **tacacs+**}
	
	Example:
	`hostname(config)# aaa-server servergroup1 protocol ldap`
	`hostname(config-aaa-server-group)`
		Creates a AAA server group.

Step 2	**tunnel-group** *groupname* **Example:** `hostname(config)# tunnel-group remotegrp`	Creates an IPsec remote access tunnel group named remotegrp.
Step 3	**tunnel-group** *groupname* **general-attributes** **Example:** `hostname(config)# tunnel-group remotegrp general-attributes`	Associates the server group and the tunnel group.
Step 4	**authorization-server-group** *group-tag* **Example:** `hostname(config-general)# authorization-server-group ldap_dir_1`	Assigns a new tunnel group to a previously created AAA server group for authorization.

Examples

While there are other authorization-related commands and options available for specific requirements, the following example shows commands for enabling user authorization with LDAP. The example then creates an IPsec remote access tunnel group named remote-1, and assigns that new tunnel group to the previously created ldap_dir_1 AAA server group for authorization:

```
hostname(config)# tunnel-group remote-1 type ipsec-ra
hostname(config)# tunnel-group remote-1 general-attributes
hostname(config-general)# authorization-server-group ldap_dir_1
hostname(config-general)#
```

After you complete this configuration work, you can then configure additional LDAP authorization parameters such as a directory password, a starting point for searching a directory, and the scope of a directory search by entering the following commands:

```
hostname(config)# aaa-server ldap_dir_1 protocol ldap
hostname(config-aaa-server-group)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# ldap-login-dn obscurepassword
hostname(config-aaa-server-host)# ldap-base-dn starthere
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)#
```

Configuring LDAP Attribute Maps

The ASA can use an LDAP directory for authenticating VPN remote access users or firewall network access/cut-thru-proxy sessions and/or for setting policy permissions (also called authorization attributes), such as ACLs, bookmark lists, DNS or WINS settings, session timers, and so on. That is, you can set the key attributes that exist in a local group policy externally through an LDAP server.

The authorization process is accomplished by means of LDAP attribute maps (similar to a RADIUS dictionary that defines vendor-specific attributes), which translate the native LDAP user attributes to Cisco ASA attribute names. You can then bind these attribute maps to LDAP servers or remove them, as needed. You can also show or clear attribute maps.

Guidelines

The ldap-attribute-map has a limitation with multi-valued attributes. For example, if a user is a memberOf of several AD groups and the ldap attribute map matches on more than one of them, the mapped value is chosen based on the alphabetization of the matched entries.

To use the attribute mapping features correctly, you need to understand Cisco LDAP attribute names and values, as well as the user-defined attribute names and values. For more information about LDAP attribute maps, see the "Active Directory/LDAP VPN Remote Access Authorization Examples" section on page C-16.

The names of frequently mapped Cisco LDAP attributes and the type of user-defined attributes that they would commonly be mapped to include the following:

- IETF-Radius-Class (Group_Policy in ASA version 8.2 and later)—Sets the group policy based on the directory's department or user group (for example, Microsoft Active Directory memberOf) attribute value. The group-policy attribute replaced the IETF-Radius-Class attribute with ASDM version 6.2/ASA version 8.2 or later.
- IETF-Radius-Filter-Id—An access control list or ACL applied to VPN clients, IPsec, and SSL.
- IETF-Radius-Framed-IP-Address—Assigns a static IP address assigned to a VPN remote access client, IPsec, and SSL.
- Banner1—Displays a text banner when the VPN remote access user logs in.
- Tunneling-Protocols—Allows or denies the VPN remote access session based on the access type.

Note A single ldapattribute map may contain one or many attributes. You can only assign one ldap attribute to a specific LDAP server.

To map LDAP features correctly, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ldap attribute-map** *map-name* **Example:** hostname(config)# ldap attribute-map att_map_1	Creates an unpopulated LDAP attribute map table.
Step 2	**map-name** *user-attribute-name Cisco-attribute-name* **Example:** hostname(config-ldap-attribute-map)# map-name department IETF-Radius-Class	Maps the user-defined attribute name department to the Cisco attribute.
Step 3	**map-value** *user-attribute-name Cisco-attribute-name* **Example:** hostname(config-ldap-attribute-map)# map-value department Engineering group1	Maps the user-defined map value department to the user-defined attribute value and the Cisco attribute value.
Step 4	**aaa-server** *server_group* [*interface_name*] **host** *server_ip* **Example:** hostname(config)# aaa-server ldap_dir_1 host 10.1.1.4	Identifies the server and the AAA server group to which it belongs.
Step 5	**ldap-attribute-map** *map-name* **Example:** hostname(config-aaa-server-host)# ldap-attribute-map att_map_1	Binds the attribute map to the LDAP server.

Examples

The following example shows how to limit management sessions to the ASA based on an LDAP attribute called accessType. The accessType attribute has three possible values:

- VPN
- admin
- helpdesk

The following example shows how each value is mapped to one of the valid IETF-Radius-Service-Type attributes that the ASA supports: remote-access (Service-Type 5) Outbound, admin (Service-Type 6) Administrative, and nas-prompt (Service-Type 7) NAS Prompt:

```
hostname(config)# ldap attribute-map MGMT
hostname(config-ldap-attribute-map)# map-name accessType IETF-Radius-Service-Type
hostname(config-ldap-attribute-map)# map-value accessType VPN 5
hostname(config-ldap-attribute-map)# map-value accessType admin 6
```

```
hostname(config-ldap-attribute-map)# map-value accessType helpdesk 7

hostname(config-ldap-attribute-map)# aaa-server LDAP protocol ldap
hostname(config-aaa-server-group)# aaa-server LDAP (inside) host 10.1.254.91
hostname(config-aaa-server-host)# ldap-base-dn CN=Users,DC=cisco,DC=local
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)# ldap-login-password test
hostname(config-aaa-server-host)# ldap-login-dn
CN=Administrator,CN=Users,DC=cisco,DC=local
hostname(config-aaa-server-host)# server-type auto-detect
hostname(config-aaa-server-host)# ldap-attribute-map MGMT
```

The following example shows how to display the complete list of Cisco LDAP attribute names:

```
hostname(config)# ldap attribute-map att_map_1
hostname(config-ldap-attribute-map)# map-name att_map_1?

ldap mode commands/options:
cisco-attribute-names:
  Access-Hours
  Allow-Network-Extension-Mode
  Auth-Service-Type
  Authenticated-User-Idle-Timeout
  Authorization-Required
  Authorization-Type
    :
    :
  X509-Cert-Data
hostname(config-ldap-attribute-map)#
```

Adding a User Account to the Local Database

This section describes how to manage users in the local database and includes the following topics:

Guidelines

The local database is used for the following features:

- 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.

Limitations

You cannot use the local database for network access authorization.

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

Detailed Steps

	Command	Purpose
Step 1	**username** *username* {**nopassword** | **password** *password* [**mschap**]} [**privilege** *priv_level*] **Example:** `hostname(config)# username exampleuser1 privilege 1`	Creates the user account. The **username** *username* keyword is a string from 4 to 64 characters long. **Note** The ASA does not prohibit the creation of usernames that only differ by case with previously configured usernames. We do not recommend this practice if VPN users are authenticated using the local user database. Usernames such as "User1" and "user1" are still distinct for authentication purposes, but if a maximum simultaneous login limit has been configured, these users share the same session count. This makes it possible for "user1" to log off "User1" by establishing a tunnel that exceeds the simultaneous login limit. The **password** *password* argument is a string from 3 to 32 characters long. The **mschap** keyword specifies that the password is converted to Unicode and hashed using MD4 after you enter it. Use this keyword if users are authenticated using MS-CHAPv1 or MS-CHAPv2. The **privilege** *level* argument 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.To limit access to privileged EXEC mode, either set the privilege level to 0 or 1, or use the **service-type** command (see Step 5). The **nopassword** keyword creates a user account with no password. The **encrypted** and **nt-encrypted** keywords are typically for display only. 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 **nt-encrypted** keyword (when you specify **mschap**). For example, if you enter the password "test," the **show running-config** output would appear as something similar to the following: `username user1 password DLaUiAX3l78qgoB5c7iVNw== nt-encrypted` The only time you would actually enter the **encrypted** or **nt-encrypted** 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.

	Command	Purpose
Step 2	**aaa authorization exec authentication-server** **Example:** `hostname(config)# aaa authorization exec authentication-server`	(Optional) Enforces user-specific access levels for users who authenticate for management access (see the **aaa authentication console LOCAL** command). This command enables management authorization for local, RADIUS, LDAP (mapped), and TACACS+ users. Use the **aaa authorization exec LOCAL** command to enable attributes to be taken from the local database. See the "Limiting User CLI and ASDM Access with Management Authorization" section on page 37-21 for information about configuring a user on a AAA server to accommodate management authorization. Note the following prerequisites for each user type: • Configure local database users at a privilege level from 0 to 15 using the **username** command. Configure the level of access using the **service-type** command. • Configure RADIUS users with Cisco VSA CVPN3000-Privilege-Level with a value between 0 and 15. • Configure LDAP users with a privilege level between 0 and 15, and then map the LDAP attribute to Cisco VAS CVPN3000-Privilege-Level using the **ldap map-attributes** command. • See the **privilege** command for information about setting command privilege levels.
Step 3	**username** *username* **attributes** **Example:** `hostname(config)# username exampleuser1 attributes`	(Optional) Configures username attributes. The *username* argument is the username that you created in Step 1.

	Command	Purpose
Step 4	**service-type** {**admin** | **nas-prompt** | **remote-access**} **Example:** `hostname(config-username)# service-type admin`	(Optional) Configures the user level if you configured management authorization in Step 2. 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 LOCAL** command, but denies ASDM configuration access if you configure the **aaa authentication http console LOCAL** command. ASDM monitoring access is allowed. If you enable authentication with the **aaa authentication enable console LOCAL** 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 LOCAL** commands (excluding the **serial** keyword; serial access is allowed). (Optional) If you are using this username for VPN authentication, you can configure many VPN attributes for the user. For more information, see the "Configuring Attributes for Specific Users" section on page 67-79.

Examples

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

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

The following example creates a user account with no password:

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

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

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

Managing User Passwords

The ASA enables administrators with the necessary privileges to modify password policy for users in the current context.

User passwords have the following guidelines:

- A maximum lifetime of 0 to 65536 days.
- A minimum length of 3 to 64 characters.
- A minimum number of changed characters for updates of 0 to 64 characters.
- They may include lower case characters.

- They may include upper case characters.
- They may include numbers.
- They may include special characters.

To specify password policy for users, perform the following steps:

	Command	Purpose
Step 1	**password-policy lifetime** *value* **Example:** `hostname (config)# password-policy lifetime 1000`	Sets the password policy for the current context and the interval in days after which passwords expire. Valid values are between 0 and 65536 days. The default value is 0 days.
Step 2	**password-policy minimum-changes** *value* **Example:** `hostname(config)# password-policy minimum-changes 4`	Sets the minimum number of characters that must be changed between new and old passwords. Valid values are between 0 and 64 characters. The default value is 0. New passwords must include a minimum of 4 character changes from the current password and are considered changed only if they do not appear anywhere in the current password.
Step 3	**password-policy minimum-length** *value* **Example:** `hostname(config)# password-policy minimum-length 8`	Sets the minimum length of passwords. Valid values are between 3 and 64 characters. The recommended minimum password length is 8 characters. If the minimum length is less than the value of any of the other minimum values (lowercase, numeric, special, and uppercase), an error message appears and the minimum length is not changed.
Step 4	**password-policy minimum-lowercase** *value* **Example:** `hostname(config)# password-policy minimum-lowercase 6`	Sets the minimum number of lower case characters that passwords may have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 5	**password-policy minimum-numeric** *value* **Example:** `hostname(config)# password-policy minimum-numeric 1`	Sets the minimum number of numeric characters that passwords may have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 6	**password-policy minimum-special** *value* **Example:** `hostname(config)# password-policy minimum-special 2`	Sets the minimum number of special characters that passwords may have. Valid values are between 0 and 64 characters. Special characters include the following: !, @, #, $, %, ^, &, *, '(‘ and ‘)’. The default value is 0, which means there is no minimum.

	Command	Purpose
Step 7	**password-policy minimum-uppercase** *value* **Example:** `hostname(config)# password-policy minimum-uppercase 3`	Sets the minimum number of upper case characters that passwords may have. Valid values are between 0 and 64 characters. The default value is 0, which means there is no minimum.
Step 8	**password-policy authenticate enable** **Example:** `hostname(config)# password-policy authenticate enable`	(Optional) Determines whether or not users are allowed to modify their own user account. If authentication is enabled, users cannot change their own password or delete their own account with the **username** command or with the **clear configure username** command.

.Changing User Passwords

The ASA enables administrators with the necessary privileges to modify passwords for users in the current context. Users must authenticate with their current passwords before they are allowed to change passwords. However, authentication is not required when an administrator is changing a user password.

To enable users to change their own account passwords, enter the following command:

Command	Purpose
change-password [**old-password** *old-password* [**new-password** *new-password*]] **Example:** `hostname# change-password old-password myoldpassword000 new password mynewpassword123`	Enables users to change their own account passwords. The **new-password** *new-password* keyword-argument pair specifies the new password. The **old-password** *old-password* keyword-argument pair specifies the old password, which reauthenticates the user. If users omit the passwords, the ASA prompts them for input. When users enter the **change-password** command, they are asked to save their running configuration.

Authenticating Users with a Public Key for SSH

Users can authenticate with a public key for SSH. The public key can be hashed or not hashed.

To authenticate with a public key for SSH, enter the following command:

Command	Purpose
username {*user*} **attributes ssh authentication publickey** *key* [**hashed**] **Example:** `hostname(config)# username anyuser ssh authentication publickey key [hashed]`	Enables public key authentication on a per-user basis. The value of the *key* argument can be one of the following: • When the *key* argument is supplied and the hashed tag is not specified, the value of the key must be a Base 64 encoded public key that is generated by SSH key generation software that can generate SSH-RSA raw keys (that is, with no certificates). After you submit the Base 64 encoded public key, that key is then hashed via SHA-256 and the corresponding 32-byte hash is used for all further comparisons. • When the *key* argument is supplied and the hashed tag is specified, the value of the key must have been previously hashed with SHA-256 and be 32 bytes long, with each byte separated by a colon (for parsing purposes). When you save the configuration, the hashed key value is saved to the configuration and used when the ASA is rebooted.

Differentiating User Roles Using AAA

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

To differentiate user roles, use the **service-type** attribute in username configuration mode. For RADIUS and LDAP (with the **ldap-attribute-map** command), you can use a Cisco Vendor-Specific Attribute (VSA), Cisco-Priv-Level, to assign a privilege level to an authenticated user.

This section includes the following topics:

Using Local Authentication

Before you configure the **service-type** attribute and privilege level when using local authentication, you must create a user, assign a password, and assign a privilege level.

To do so, enter the following command:

```
hostname(config)# username admin password mysecret123 privilege 15
```

Where **mysecret123** is the stored password and 15 is the assigned privilege level, which indicates an admin user.

The available configuration options for the **service-type** attribute include the following:

- **admin**, in which users are allowed access to the configuration mode. This option also allows a user to connect via remote access.
- **nas-prompt**, in which users are allowed access to the EXEC mode.
- **remote-access**, in which users are allowed access to the network.

The following example designates a **service-type** of **admin** for a user named admin:

```
hostname(config)# username admin attributes
hostname(config-username)# service-type admin
```

The following example designates a **service-type** of **remote-access** for a user named ra-user:

```
hostname(config)# username ra-user attributes
hostname(config-username)# service-type remote-access
```

Using RADIUS Authentication

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 supported attribute values are the following: administrative(6), nas-prompt(7), Framed(2), and Login(1). For a list of supported RADIUS IETF VSAs used for authentication and authorization, see Table C-8 on page C-36.

For more information about using RADIUS authentication, see "Configuring an External RADIUS Server" section on page C-27. For more information about configuring RADIUS authentication for Cisco Secure ACS, see the Cisco Secure ACS documentation on Cisco.com.

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. For a list of supported RADIUS VSAs used for authorization, see Table C-7 on page C-28.

Using LDAP Authentication

When users are authenticated through LDAP, the native LDAP attributes and their values can be mapped to Cisco ASA attributes to provide specific authorization features. For the supported list of LDAP VSAs used for authorization, see Table C-2 on page C-6.

You can use the LDAP attribute mapping feature for LDAP authorization. For examples of this feature, see the "Understanding Policy Enforcement of Permissions and Attributes" section on page C-1.

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.

To define an LDAP attribute map, enter the following commands:

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

The following is sample output from the **ldap-attribute-map** command:

```
ldap attribute-map admin-control
```

```
map-name company Privilege-Level
map-name title IETF-Radius-Service-Type
```

To apply the LDAP attribute map to the LDAP AAA server, enter the following commands:

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

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

Using TACACS+ Authentication

For information about how to configure TACACS+ authentication, see the "RADIUS Accounting Disconnect Reason Codes" section on page C-37.

Monitoring AAA Servers

To monitor AAA servers,enter one of the following commands:

Command	Purpose
show aaa-server	Shows the configured AAA server 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.
show running-config all ldap attribute-map	Shows all LDAP attribute maps in the running configuration. To clear all LDAP attribute maps in the running configuration, use the **clear configuration ldap attribute-map** command.
show running-config zonelabs-integrity	Shows the Zone Labs Integrity server configuration. To clear the Zone Labs Integrity server configuration, use the **clear configure zonelabs-integrity** command.
show ad-groups *name* [**filter** *string*]	Applies only to AD servers using LDAP, and shows groups that are listed on an AD server.
show running-config [all] password-policy	Shows the password policy for the current context.

Additional References

For additional information related to implementing LDAP mapping, see the "RFCs" section on page 35-31.

RFCs

RFC	Title
2138	*Remote Authentication Dial In User Service (RADIUS)*
2139	*RADIUS Accounting*
2548	*Microsoft Vendor-specific RADIUS Attributes*
2868	*RADIUS Attributes for Tunnel Protocol Support*

Feature History for AAA Servers

Table 35-3 lists each feature change and the platform release in which it was implemented.

Table 35-3 Feature History for AAA Servers

Feature Name	Platform Releases	Feature Information
AAA Servers	7.0(1)	AAA Servers describe support for AAA and how to configure AAA servers and the local database. We introduced the following commands: **username**, **aaa authorization exec authentication-server**, **aaa authentication console LOCAL**, **aaa authorization exec LOCAL**, **service-type, ldap attribute-map**, **aaa-server protocol**, **aaa authentication** {**telnet** | **ssh** | **serial**} **console LOCAL**, **aaa authentication http console LOCAL, aaa authentication enable console LOCAL, max-failed-attempts, reactivation-mode, accounting-mode simultaneous, aaa-server host, authorization-server-group, tunnel-group, tunnel-group general-attributes, map-name, map-value, ldap-attribute-map, zonelabs-Integrity server-address**, **zonelabs-integrity port**, **zonelabs-integrity interface**, **zonelabs-integrity fail-timeout**, **zonelabs-integrity fail-close**, **zonelabs-integrity fail-open**, **zonelabs-integrity ssl-certificate-port**, **zonelabs-integrity ssl-client-authentication {enable | disable}**, **client-firewall {opt | req} zonelabs-integrity**
Key vendor-specific attributes (VSAs) sent in RADIUS access request and accounting request packets from the ASA	8.4(3)	Four New VSAs—Tunnel Group Name (146) and Client Type (150) are sent in RADIUS access request packets from the ASA. Session Type (151) and Session Subtype (152) are sent in RADIUS accounting request packets from the ASA. All four attributes are sent for all accounting request packet types: Start, Interim-Update, and Stop. The RADIUS server (for example, ACS and ISE) can then enforce authorization and policy attributes or use them for accounting and billing purposes.
Common Criteria certification and FIPS support for password policy, password change, and SSH public key authentication	8.4(4.1)	We introduced or modified the following commands: **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**, **username**, **username attributes**, **clear configure username**, **change-password**, **clear configure password-policy**, **show running-config password-policy**, and **username**.