



CHAPTER 13

Configuring AAA Servers and the Local Database

This chapter describes support for AAA (pronounced “triple A”) and how to configure AAA servers and the local database.

This chapter contains the following sections:

- [AAA Overview, page 13-1](#)
- [AAA Server and Local Database Support, page 13-2](#)
- [Configuring the Local Database, page 13-10](#)
- [Identifying AAA Server Groups and Servers, page 13-12](#)
- [Using Certificates and User Login Credentials, page 13-15](#)
- [Supporting a Zone Labs Integrity Server, page 13-16](#)

AAA Overview

AAA enables the security appliance 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 make it through the security appliance. (The Telnet server enforces authentication, too; the security appliance 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:

- [About Authentication, page 13-1](#)
- [About Authorization, page 13-2](#)
- [About Accounting, page 13-2](#)

About Authentication

Authentication controls access by requiring valid user credentials, which are typically a username and password. You can configure the security appliance to authenticate the following items:

- All administrative connections to the security appliance including the following sessions:
 - Telnet
 - SSH
 - Serial console
 - ASDM (using HTTPS)
 - VPN management access
- The **enable** command
- Network access
- VPN access

About Authorization

Authorization controls access *per user* after users authenticate. You can configure the security appliance to authorize the following items:

- Management commands
- Network access
- VPN access

Authorization controls the services and commands available to each authenticated user. Were you not to 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 authenticate inside users who attempt to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

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

About Accounting

Accounting tracks traffic that passes through the security appliance, 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 when sessions start and stop, username, the number of bytes that pass through the security appliance for the session, the service used, and the duration of each session.

AAA Server and Local Database Support

The security appliance supports a variety of AAA server types and a local database that is stored on the security appliance. This section describes support for each AAA server type and the local database.

This section contains the following topics:

- [Summary of Support, page 13-3](#)

- [RADIUS Server Support, page 13-3](#)
- [TACACS+ Server Support, page 13-4](#)
- [SDI Server Support, page 13-4](#)
- [NT Server Support, page 13-5](#)
- [Kerberos Server Support, page 13-5](#)
- [LDAP Server Support, page 13-6](#)
- [SSO Support for WebVPN with HTTP Forms, page 13-9](#)
- [Local Database Support, page 13-9](#)

Summary of Support

Table 13-1 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, refer to the topics following the table.

Table 13-1 Summary of AAA Support

AAA Service	Database Type							
	Local	RADIUS	TACACS+	SDI	NT	Kerberos	LDAP	HTTP Form
Authentication of...								
VPN users	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes ¹
Firewall sessions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Administrators	Yes	Yes	Yes	Yes ²	Yes	Yes	Yes	No
Authorization of...								
VPN users	Yes	Yes	No	No	No	No	Yes	No
Firewall sessions	No	Yes ³	Yes	No	No	No	No	No
Administrators	Yes ⁴	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 ⁵	Yes	No	No	No	No	No

1. HTTP Form protocol supports single sign-on authentication for WebVPN users only.

2. SDI is not supported for HTTP administrative access.

3. For firewall sessions, RADIUS authorization is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response.

4. Local command authorization is supported by privilege level only.

5. Command accounting is available for TACACS+ only.

RADIUS Server Support

The security appliance supports RADIUS servers.

This section contains the following topics:

- [Authentication Methods, page 13-4](#)
- [Attribute Support, page 13-4](#)
- [RADIUS Authorization Functions, page 13-4](#)

Authentication Methods

The security appliance supports the following authentication methods with RADIUS:

- PAP—For all connection types.
- CHAP—For L2TP-over-IPSec.
- MS-CHAPv1—For L2TP-over-IPSec.
- MS-CHAPv2—For L2TP-over-IPSec, and for regular IPSec remote access connections when the password management feature is enabled.

Attribute Support

The security appliance 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 VSAs, identified by RADIUS vendor ID 9.
- Cisco VPN-related VSAs, identified by RADIUS vendor ID 3076.
- Microsoft VSAs, defined in RFC 2548.

RADIUS Authorization Functions

The security appliance can use RADIUS servers for user authorization for network access 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 security appliance. Access to a given service is either permitted or denied by the access list. The security appliance deletes the access list when the authentication session expires.

TACACS+ Server Support

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

SDI Server Support

The RSA SecureID servers are also known as SDI servers.

This section contains the following topics:

- [SDI Version Support, page 13-5](#)

- [Two-step Authentication Process, page 13-5](#)
- [SDI Primary and Replica Servers, page 13-5](#)

SDI Version Support

The security appliance supports SDI Version 5.0 and 6.0. 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/Server IP address with .sdi appended.

A version 5.0 or 6.0 SDI server that you configure on the security appliance can be either the primary or any one of the replicas. See the “[SDI Primary and Replica Servers](#)” section on [page 13-5](#) for information about how the SDI agent selects servers to authenticate users.

Two-step Authentication Process

SDI version 5.0 and 6.0 uses 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 means that the same user cannot authenticate to two security appliances using the same authentication servers simultaneously. After a successful username lock, the security appliance sends the passcode.

SDI Primary and Replica Servers

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

NT Server Support

The security appliance 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. This is a limitation of NTLM version 1.

Kerberos Server Support

The security appliance supports 3DES, DES, and RC4 encryption types.

**Note**

The security appliance 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 security appliance.

For a simple Kerberos server configuration example, see [Example 13-2](#).

LDAP Server Support

This section describes using an LDAP directory with the security appliance for user authentication and VPN authorization. This section includes the following topics:

- [Authentication with LDAP, page 13-6](#)
- [Authorization with LDAP for VPN, page 13-7](#)
- [LDAP Attribute Mapping, page 13-8](#)

For example configuration procedures used to set up LDAP authentication or authorization, see [Appendix E, “Configuring an External Server for Authorization and Authentication”](#).

Authentication with LDAP

During authentication, the security appliance acts as a client proxy to the LDAP server for the user, and authenticates to the LDAP server in either plain text or using the Simple Authentication and Security Layer (SASL) protocol. By default, the security appliance passes authentication parameters, usually a username and password, to the LDAP server in plain text. Whether using SASL or plain text, you can secure the communications between the security appliance and the LDAP server with SSL using the **ldap-over-ssl** command.



Note

If you do not configure SASL, we strongly recommend that you secure LDAP communications with SSL. See the **ldap-over-ssl** command in the *Cisco Security Appliance Command Reference*.

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 which is applied to the VPN session. Thus, using LDAP accomplishes authentication and authorization in a single step.

Securing LDAP Authentication with SASL

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

- Digest-MD5 — The security appliance responds to the LDAP server with an MD5 value computed from the username and password.
- Kerberos — The security appliance responds to the LDAP server by sending the username and realm using the GSSAPI (Generic Security Services Application Programming Interface) Kerberos mechanism.

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

The following example configures the security appliance for authentication to an LDAP directory server named `ldap_dir_1` using the digest-MD5 SASL mechanism, and communicating over an SSL-secured connection:

```
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)# sasl-mechanism digest-md5
hostname(config-aaa-server-host)# ldap-over-ssl enable
```

```
hostname(config-aaa-server-host)#
```

Setting the LDAP Server Type

The security appliance supports LDAP Version 3. In the current release, it is compatible only with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory. In later releases, the security appliance will support other OpenLDAP servers.

By default, the security appliance auto-detects whether it is connected to a Microsoft or a Sun LDAP directory server. However, if auto-detection fails to determine the LDAP server type, and you know the server is either a Microsoft or Sun server, you can manually configure the server type. The following example sets the LDAP directory server `ldap_dir_1` to the Sun Microsystems type:

```
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)# server-type sun
hostname(config-aaa-server-host)#
```



Note

- Sun—The DN configured on the security appliance 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 ACI on the default password policy.
- Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

Authorization with LDAP for VPN

When user LDAP authentication for VPN access has succeeded, the security appliance 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, you must first create a AAA server group and a tunnel group. You then associate the server and tunnel groups using the **tunnel-group general-attributes** command. While there are other authorization-related commands and options available for specific requirements, the following example shows fundamental commands for enabling user authorization with LDAP. This 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 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 fundamental configuration work, you can configure additional LDAP authorization parameters such as a directory password, a starting point for searching a directory, and the scope of a directory search:

```
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)#
```

See LDAP commands in the *Cisco Security Appliance Command Reference* for more information.

LDAP Attribute Mapping

If you are introducing a security appliance to an existing LDAP directory, your existing LDAP attribute names and values are probably different from the existing ones. You must create LDAP attribute maps that map your existing user-defined attribute names and values to Cisco attribute names and values that are compatible with the security appliance. You can then bind these attribute maps to LDAP servers or remove them as needed. You can also show or clear attribute maps.



Note

To use the attribute mapping features correctly, you need to understand the Cisco LDAP attribute names and values as well as the user-defined attribute names and values.

The following command, entered in global configuration mode, creates an unpopulated LDAP attribute map table named att_map_1:

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

The following commands map the user-defined attribute name department to the Cisco attribute name cVPN3000-IETF-Radius-Class. The second command maps the user-defined attribute value Engineering to the user-defined attribute department and the Cisco-defined attribute value group1.

```
hostname(config)# ldap attribute-map att_map_1
hostname(config-ldap-attribute-map)# map-name department cVPN3000-IETF-Radius-Class
hostname(config-ldap-attribute-map)# map-value department Engineering group1
hostname(config-ldap-attribute-map)#
```

The following commands bind the attribute map att_map_1 to the LDAP server ldap_dir_1:

```
hostname(config)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# ldap-attribute-map att_map_1
hostname(config-aaa-server-host)#
```



Note

The command to create an attribute map (**ldap attribute-map**) and the command to bind it to an LDAP server (**ldap-attribute-map**) differ only by a hyphen and the mode.

The following commands display or clear all LDAP attribute maps in the running configuration:

```
hostname# show running-config all ldap attribute-map
hostname(config)# clear configuration ldap attribute-map
hostname(config)#
```

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

```

cVPN3000-IETF-Radius-Class - Department or user group
cVPN3000-IETF-Radius-Filter-Id - Access control list
cVPN3000-IETF-Radius-Framed-IP-Address - A static IP address
cVPN3000-IPSec-Banner1 - A organization title
cVPN3000-Tunneling-Protocols - Allow or deny dial-in

```

For a list of Cisco LDAP attribute names and values, see [Appendix E, “Configuring an External Server for Authorization and Authentication”](#). Alternatively, you can enter “?” within ldap-attribute-map mode to display the complete list of Cisco LDAP attribute names, as shown in the following example:

```

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:
  cVPN3000-Access-Hours
  cVPN3000-Allow-Network-Extension-Mode
  cVPN3000-Auth-Service-Type
  cVPN3000-Authenticated-User-Idle-Timeout
  cVPN3000-Authorization-Required
  cVPN3000-Authorization-Type
  :
  :
  cVPN3000-X509-Cert-Data
hostname(config-ldap-attribute-map)#

```

SSO Support for WebVPN with HTTP Forms

The security appliance can use the HTTP Form protocol for single sign-on (SSO) authentication of WebVPN users only. Single sign-on support lets WebVPN users enter a username and password only once to access multiple protected services and Web servers. The WebVPN server running on the security appliance acts as a proxy for the user to the authenticating server. When a user logs in, the WebVPN server sends an SSO authentication request, including username and password, to the authenticating server using HTTPS. If the server approves the authentication request, it returns an SSO authentication cookie to the WebVPN server. The security appliance keeps this cookie on behalf of the user and uses it to authenticate the user to secure websites within the domain protected by the SSO server.

In addition to the HTTP Form protocol, WebVPN administrators can choose to configure SSO with the HTTP Basic and NTLM authentication protocols (the **auto-signon** command), or with Computer Associates eTrust SiteMinder SSO server (formerly Netegrity SiteMinder) as well. For an in-depth discussion of configuring SSO with either HTTP Forms, **auto-signon** or SiteMinder, see the [Configuring WebVPN](#) chapter.

Local Database Support

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

This section contains the following topics:

- [User Profiles, page 13-10](#)
- [Fallback Support, page 13-10](#)

User Profiles

User profiles contain, at a minimum, a username. Typically, a password is assigned to each username, although passwords are optional.

The **username attributes** command lets you enter the username mode. In this mode, you can add other information to a specific user profile. The information you can add includes VPN-related attributes, such as a VPN session timeout value.

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 security appliance.

For users who need fallback support, we recommend that their usernames and passwords in the local database match their usernames and passwords in the AAA servers. This 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**—When you use the **aaa authentication console** command, you can add the **LOCAL** keyword after the AAA server group tag. If the servers in the group all are unavailable, the security appliance uses the local database to authenticate administrative access. This can include enable password authentication, too.
- **Command authorization**—When you use the **aaa authorization command** command, you can add the **LOCAL** keyword after the AAA server group tag. If the TACACS+ servers in the group all are 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 security appliance if AAA servers that normally support these VPN services are unavailable. The **authentication-server-group** command, available in tunnel-group general attributes mode, lets you specify the **LOCAL** keyword when you are configuring attributes of a tunnel group. When 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.

Configuring the Local Database

This section describes how to manage users in the local database. You can use the local database for CLI access authentication, privileged mode authentication, command authorization, network access authentication, and VPN authentication and authorization. You cannot use the local database for network access authorization. The local database does not support accounting.

For multiple context mode, you can configure usernames in the system execution space to provide individual logins using the **login** command; however, you cannot configure any **aaa** commands in the system execution space.



Caution

If you add to the local database users who can gain access to the CLI but who should not be allowed to enter privileged mode, enable command authorization. (See the [“Configuring Local Command Authorization”](#) section on page 40-8.) Without command authorization, users can access privileged

mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication so that the user cannot use the **login** command, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

To define a user account in the local database, perform the following steps:

Step 1 Create the user account. To do so, enter the following command:

```
hostname(config)# username name {nopassword | password password [mschap]} [privilege
priv_level]
```

where the options are as follows:

- **username**—A string from 4 to 64 characters long.
- **password** *password*—A string from 3 to 16 characters long.
- **mschap**—Specifies that the password will be converted to unicode and hashed using MD4 after you enter it. Use this keyword if users are authenticated using MSCHAPv1 or MSCHAPv2.
- **privilege level**—The privilege level that you want to assign to the new user account (from 0 to 15). The default is 2. This privilege level is used with command authorization.
- **nopassword**—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 security appliance 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** display would appear to be something like the following:

```
username pat password DLaUiAX3178qgoB5c7iVNw== 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 to another security appliance and you are using the same password.

Step 2 To configure a local user account with VPN attributes, follow these steps:

a. Enter the following command:

```
hostname(config)# username username attributes
```

When you enter a **username attributes** command, you enter username mode. The commands available in this mode are as follows:

- **group-lock**
- **password-storage**
- **vpn-access-hours**
- **vpn-filter**
- **vpn-framed-ip-address**
- **vpn-group-policy**
- **vpn-idle-timeout**
- **vpn-session-timeout**
- **vpn-simultaneous-logins**
- **vpn-tunnel-protocol**

- **webvpn**

Use these commands as needed to configure the user profile. For more information about these commands, see the *Cisco Security Appliance Command Reference*.

- When you have finished configuring the user profiles, enter **exit** to return to config mode.

For example, the following command assigns a privilege level of 15 to the admin user account:

```
hostname(config)# username admin password passw0rd privilege 15
```

The following command creates a user account with no password:

```
hostname(config)# username bcham34 nopassword
```

The following commands create a user account with a password, enter username mode, and specify a few VPN attributes:

```
hostname(config)# username rwilliams password g0ge0us
hostname(config)# username rwilliams attributes
hostname(config-username)# vpn-tunnel-protocol IPSec
hostname(config-username)# vpn-simultaneous-logins 6
hostname(config-username)# exit
```

Identifying AAA Server Groups and Servers

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+.

The security appliance contacts the first server in the group. If that server is unavailable, the security appliance contacts the next server in the group, if configured. If all servers in the group are unavailable, the security appliance 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 security appliance continues to try the AAA servers.

To create a server group and add AAA servers to it, follow these steps:

Step 1 For each AAA server group you need to create, follow these steps:

- Identify the server group name and the protocol. To do so, enter the following command:

```
hostname(config)# aaa-server server_group protocol {kerberos | ldap | nt | radius |
sdi | tacacs+}
```

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 15 single-mode server groups or 4 multi-mode server groups. Each server group can have up to 16 servers in single mode or up to 4 servers in multi-mode.

When you enter a **aaa-server protocol** command, you enter group mode.

- If you want to specify the maximum number of requests sent to a AAA server in the group before trying the next server, enter the following command:

```
hostname(config-aaa-server-group)# max-failed-attempts number
```

The *number* can be between 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only; see the “Configuring AAA for System Administrators” section on page 40-5 and the “Configuring TACACS+ Command Authorization” section on page 40-11 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 following step.

If you do not have a fallback method, the security appliance continues to retry the servers in the group.

- c. If you want to specify the method (reactivation policy) by which failed servers in a group are reactivated, enter the following command:

```
hostname(config-aaa-server-group)# # reactivation-mode {depletion [deadtime minutes] | timed}
```

Where the **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive.

The **deadtime** *minutes* argument 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 re-enabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

- d. If you want to send accounting messages to all servers in the group (RADIUS or TACACS+ only), enter the following command:

```
hostname(config-aaa-server-group)# accounting-mode simultaneous
```

To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.

Step 2 For each AAA server on your network, follow these steps:

- a. Identify the server, including the AAA server group it belongs to. To do so, enter the following command:

```
hostname(config)# aaa-server server_group (interface_name) host server_ip
```

When you enter a **aaa-server host** command, you enter host mode.

- b. As needed, use host mode commands to further configure the AAA server.

The commands in host mode do not apply to all AAA server types. Table 13-2 lists the available commands, the server types they apply to, and whether a new AAA server definition has a default value for that command. Where a command is applicable to the server type you specified and no default value is provided (indicated by “—”), use the command to specify the value. For more information about these commands, see the *Cisco Security Appliance Command Reference*.

Table 13-2 Host Mode Commands, Server Types, and Defaults

Command	Applicable AAA Server Types	Default Value
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	—
ldap-login-password	LDAP	—
ldap-naming-attribute	LDAP	—
ldap-over-ssl	LDAP	—
ldap-scope	LDAP	—
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
timeout	All	10 seconds

[Example 13-1](#) shows commands that add one TACACS+ group with one primary and one backup server, one RADIUS group with a single server, and an NT domain server.

Example 13-1 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 NTAAuth protocol nt
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server NTAAuth (inside) host 10.1.1.4
hostname(config-aaa-server-host)# nt-auth-domain-controller primary1
hostname(config-aaa-server-host)# exit

```

Example 13-2 shows commands that 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 13-2** does not define a retry interval or the port that the Kerberos server listens to, the security appliance uses the default values for these two server-specific parameters. **Table 13-2** lists the default values for all AAA server host mode commands.


Note

Kerberos realm names use numbers and upper-case letters only. Although the security appliance 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 13-2 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)#

```

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. This applies to both IPsec and WebVPN.

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.

Using User Login Credentials

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

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

Using certificates

If user digital certificates are configured, the security appliance first validates the certificate. It does not, however, use any of the DN's from the certificates as a username for the authentication.

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

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

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

- Authentication
 - DISABLED (set to None) by authentication server group setting
 - No credentials used
- Authorization
 - Enabled by 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 security appliance uses the secondary DN field value as the username for the authorization request.

For example, consider a user certificate that contains 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.

Supporting a Zone Labs Integrity Server

This section introduces the Zone Labs Integrity Server, also called Check Point Integrity Server, and presents an example procedure for configuring the security appliance to support the Zone Labs Integrity Server. The Integrity server is a central management station for configuring and enforcing security policies on remote PCs. If a remote PC does not conform to the security policy dictated by the Integrity Server, it will not be granted access to the private network protected by the Integrity Server and security appliance.

This section includes the following topics:

- [Overview of Integrity Server and Security Appliance Interaction, page 13-17](#)
- [Configuring Integrity Server Support, page 13-17](#)

Overview of Integrity Server and Security Appliance Interaction

The VPN client software and the Integrity client software are co-resident on a remote PC. The following steps summarize the actions of the remote PC, security appliance, and Integrity server in the establishment of a session between the PC and the enterprise private network:

1. The VPN client software (residing on the same remote PC as the Integrity client software) connects to the security appliance and tells the security appliance what type of firewall client it is.
2. Once it approves the client firewall type, the security appliance passes Integrity server address information back to the Integrity client.
3. With the security appliance acting as a proxy, the Integrity client establishes a restricted connection with the Integrity server. A restricted connection is only between the Integrity client and server.
4. The Integrity server determines if the Integrity client is in compliance with the mandated security policies. If the client is in compliance with security policies, the Integrity server instructs the security appliance to open the connection and provide the client with connection details.
5. On the remote PC, the VPN client passes connection details to the Integrity client and signals that policy enforcement should begin immediately and the client can no longer enter the private network.
6. Once the connection is established, the server continues to monitor the state of the client using client heartbeat messages.

**Note**

The current release of the security appliance supports one Integrity Server at a time even though the user interfaces support the configuration of up to five Integrity Servers. If the active Server fails, configure another Integrity Server on the security appliance and then reestablish the client VPN session.

Configuring Integrity Server Support

This section describes an example procedure for configuring the security appliance to support the Zone Labs Integrity Servers. The procedure involves configuring address, port, connection fail timeout and fail states, and SSL certificate parameters.

First, you must configure the hostname or IP address of the Integrity server. The following example commands, entered in global configuration mode, configure an Integrity server using the IP address 10.0.0.5. They also specify port 300 (the default port is 5054) and the inside interface for communications with the Integrity server.

```
hostname(config)# zonelabs-integrity server-address 10.0.0.5
hostname(config)# zonelabs-integrity port 300
hostname(config)# zonelabs-integrity interface inside
hostname(config)#
```

If the connection between the security appliance and the Integrity server fails, the VPN client connections remain open by default so that the enterprise VPN is not disrupted by the failure of an Integrity server. However, you may want to close the VPN connections if the Zone Labs Integrity Server fails. The following commands ensure that the security appliance waits 12 seconds for a response from either the active or standby Integrity servers before declaring an the Integrity server as failed and closing the VPN client connections:

```
hostname(config)# zonelabs-integrity fail-timeout 12
hostname(config)# zonelabs-integrity fail-close
hostname(config)#
```

The following command returns the configured VPN client connection fail state to the default and ensures the client connections remain open:

```
hostname(config)# zonelabs-integrity fail-open
hostname(config)#
```

The following example commands specify that the Integrity server connects to port 300 (default is port 80) on the security appliance to request the server SSL certificate. While the server SSL certificate is always authenticated, these commands also specify that the client SSL certificate of the Integrity server be authenticated.

```
hostname(config)# zonelabs-integrity ssl-certificate-port 300
hostname(config)# zonelabs-integrity ssl-client-authentication
hostname(config)#
```

To set the firewall client type to the Zone Labs Integrity type, use the **client-firewall** command as described in the [“Configuring Firewall Policies” section on page 30-54](#). The command arguments that specify firewall policies are not used when the firewall type is **zonelabs-integrity** because the Integrity server determines the policies.