CHAPTER 2

Configuring Authentication and Accounting Services

This chapter describes how to configure the Cisco 4700 Series Application Control Engine (ACE) appliance to perform user authentication and accounting (AAA) services to provide a higher level of security for users accessing the ACE. The AAA services allow you to use multiple AAA servers to control who can access the ACE and to track the actions of each user who accesses the ACE. Based on the username and password combination provided, the ACE performs local user authentication using the local database or remote user authentication and accounting using external AAA servers.

This chapter contains the following major sections:

- AAA Overview
- Authentication and Accounting Configuration Quick Start
- Configuring the AAA Server
- Creating User Accounts
- Configuring the ACE as a Client of a RADIUS, TACACS+, or LDAP Server
- Defining the Login Authentication Method
- Defining the Default Accounting Method
- Viewing AAA Status and Statistics

AAA Overview

AAA provides management security for user access to the ACE through a combination of authentication and accounting services. AAA informs the ACE who the user is and what the user did. You can use authentication alone or with accounting. ACE provides security for the management access methods to the ACE, including the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

You can access the ACE CLI through the console port or by a Telnet or SSH session. When you log in to the ACE using either a Telnet or SSH connection, and if the ACE is configured for AAA server-based authentication, a temporary SNMP user entry is automatically created. The SNMPv3 protocol data units (PDUs) with the associated Telnet or SSH login name as the SNMPv3 user are authenticated by the ACE.

As part of the authentication process, the ACE associates each user with a user role and a domain privilege pair under a specific virtual context. Each virtual context behaves like an independent device with its own configuration, security policies, interfaces, and domains. A user context can be independently managed with other user contexts. A domain provides a namespace in which a user operates, and each user is associated with at least one domain. The role assigned to a user determines the operations that a user can perform on the objects in a domain and the command set available to that user. Each context has a virtual AAA instance running to provide authentication for the users logging in and accounting services to log user activity.

Each virtual context on the ACE can have its own IP address. You can access each virtual context in an ACE through the console port or a Telnet or SSH session by specifying this IP address. Users can also send SNMP requests to the ACE by using this IP address.

Note Only the Admin context is accessible through the console port; all other contexts can be reached through Telnet or SSH.

The administrator of each virtual context is able to perform, independent from other contexts, the following actions:

- Configure different AAA servers and their parameters

- Create the same username across contexts, and associate the username with a unique role in a context and multiple domains
- Share AAA servers. Each user, however, must be authenticated for each virtual context and must use the same password
- Log user accounting activities, which are distinguished by the context in which a user has signed in
- Display the users currently authenticated on the virtual context

Each user who accesses the ACE from a specific IP address needs to authenticate once only. The user authentication sequence remains in effect until the authentication session expires on the ACE.

The ACE runs the AAA client, which sits between the users and the AAA server. On one side, the ACE prompts each user for their credentials (username and password). On the other side, the ACE queries the identified AAA servers to determine if the user being authenticated has supplied the correct credentials and is authorized access to the ACE.

The ACE performs authentication using either the local user database that resides on the ACE or a remote AAA server. The ACE can use a Remote Access Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), or Lightweight Directory Access Protocol (v3) (LDAP) server for remote authentication and designation of access rights.

This section contains the following topics:

- Local Database and Remote Server Support
- Authentication Overview
- Accounting Overview

Local Database and Remote Server Support

The ACE supports local authentication using a local database on the ACE or remote authentication using one or more AAA servers. AAA remote servers are grouped into independent groups of TACACS+, RADIUS, or LDAP servers. For a group of servers, the ACE bases the selection of the server to use on the first active server in the group.

Note "First" refers to the order in which servers have been configured.

When a user logs in to an ACE, the servers are accessed one at a time, starting with the first server specified in the configuration, until a server responds to the ACE.

When you configure server groups using the server group authentication method, the ACE sends an authentication request to the first AAA server in the group as follows:

- If the remote AAA server fails to respond, the ACE attempts to contact the next server in the group until a remote AAA server responds to the authentication request.
- If all AAA servers in the server group fail to respond, the ACE tries to contact the AAA servers in the next configured server group.
- If all remote AAA servers fail to respond, by default, the ACE attempts to authenticate the user against the local database.

If the username and password are successfully authenticated either locally or remotely, the ACE allows the user to log in, and the user is assigned a unique role (as specified through the **role** command, which determines the commands and resources available to each user).

Each server within a group can assume an active or an inactive state if a network connection failure occurs. The policy used to select the AAA server takes the server state into account. The ACE monitors the AAA server operation by sending authentication requests to a timed-out server. If the ACE does not receive confirmation from the server within a user-specified number of retries, the ACE declares the server to be unresponsive and initiates the sequence to contact the next available server specified in the server group.

If a dead-time interval is specified for a AAA server and the connection to server A fails, the ACE marks server A as out of service and skips server A for the duration of the dead-time interval. The ACE then sends probe access-request packets to verify that the AAA server is available and can receive authentication requests. When the server responds to a probe access-request packet, the connection resumes to server A.

This section contains the following topics:

- Local Database
- TACACS+ Server
- RADIUS Server
- LDAP Directory Server

Local Database

You can configure user account access to the local database on the ACE for CLI access authentication. When a user attempts to access the ACE CLI by using the console port or a Telnet or SSH session, the ACE consults the local user database for the username and password. By default, each user assumes the Network-Monitor role and is allowed to operate on all domains.

If you specify local authentication as the fallback method and the specified AAA servers in a server group are unavailable for authentication, the ACE then attempts to access the local database to perform user authentication and accounting.

TACACS+ Server

TACACS+ controls user access to the ACE by exchanging Network Access Server (NAS) information between the ACE and a centralized database to determine the identity of a user. TACACS+ is an enhanced version of TACACS, a User Datagram Protocol (UDP)-based access-control protocol that is specified by RFC 1492. TACACS+ uses TCP to ensure reliable delivery and encrypt all traffic between the TACACS+ server and the TACACS+ daemon on the ACE.

A TACACS+ server can provide user authentication and accounting functions. These services, while all part of TACACS+, are independent of one another, so a given TACACS+ configuration can use any or all of the services.

The TACACS+ protocol encrypts the user password information using the MD5 encryption algorithm and adds a TACACS+ packet header. This header information identifies the packet type being sent (for example, an authentication packet), the packet sequence number, the encryption type being used, and the total packet length. The TACACS+ protocol forwards the packet to the TACACS+ server.

To maintain security between the ACE and the TACACS+ server, you can specify an encryption key (shared secret) for all communication between the ACE and the TACACS+ server. For correct operation, you must specify the identical encryption key on both the ACE and the TACACS+ server.

RADIUS Server

RADIUS is a client-server access protocol that is used by the NAS to authenticate users attempting to connect to the ACE. The NAS functions as a client, passing user information to one or more RADIUS servers. The NAS permits or denies

network access to a user based on the response that it receives from a RADIUS server. RADIUS uses UDP for connectionless transport between the RADIUS client and server. For more information about how the RADIUS protocol operates, see RFC 2138.

To maintain security between the ACE and the RADIUS server, you can specify an encryption key (shared secret) for all communication between the ACE and the RADIUS server. For correct operation, you must specify the identical encryption key on both the ACE and the RADIUS server.

LDAP Directory Server

LDAP is an open-standard client-server authentication protocol for accessing X.500 Directory Access Protocol (DAP) directory services. LDAP runs over TCP/IP or other connection-oriented transfer services. The ACE supports only LDAP version 3 for simple authentication and search operations. For more information about how the LDAP protocol operates, see RFC 2251.

The LDAP information model is based on entries. An entry is a collection of attributes that has a globally unique distinguished name (DN). The DN is used in the LDAP database to refer to an entry. Each entry contains one or more attributes that describe the entry, and each attribute has a type and one or more values. The types are mnemonic strings, such as “cn” for a common name, or “mail” for an e-mail address.

The LDAP client (the ACE) requests user authentication with the LDAP server and retrieves the user profile by requesting a search through the directory database maintained by the server. The LDAP server maintains a directory of entries, which are arranged into a hierarchical structure called the Directory Information Tree (DIT).

The LDAP client performs operations on the directory data. LDAP allows you to search the directory for data that meets the arbitrary user-specified criteria. You can specify which part of the directory to search and what information to return. A search filter that uses Boolean conditions specifies the directory data that matches the search.

Note The ACE does not support update, compare, and cancel operations with the LDAP server. In addition, the ACE does not support an unsolicited notification from the LDAP server. Supported messages include bindRequest, bindResponse, unbindRequest, searchRequest, searchResEntry, and searchResDone.

Authentication Overview

Authentication allows you to control user access to the ACE CLI by requiring specification of a valid username and password. You can access the ACE CLI through the console port or by a Telnet or SSH session. For each management access path to the ACE, you can configure one or more of the following security control options: local database, remote (RADIUS, TACACS+, or LDAP), or no password verification.

The host is prompted by the ACE to provide a valid username and password. After the designated RADIUS, TACACS+, or LDAP server authenticates the username and password, the ACE provides access rights to the user.

Accounting Overview

Accounting tracks and maintains a log of useful information during each user management session with the ACE. You can use this information to generate reports for troubleshooting and auditing purposes. Accounting logs can be stored locally on the ACE or sent to remote AAA servers. If the ACE is also configured to authenticate the user, the AAA server maintains accounting information by username.

- For a TACACS+ server, accounting information includes user commands entered on the ACE, the duration of each session, and when sessions start and stop.
- For a RADIUS server, accounting information includes the duration of each session and when sessions start and stop.

When ACE user commands are being logged on a TACACS+ server, the server prefaces each command with either a “<0:>” or a “<1:>” to indicate the success or failure of the command as issued by a user on the ACE CLI. For example, when a user attempts to enter a command on the ACE and that user does not have the correct role privileges, a “<1:>” appears to indicate a failure.

Authentication and Accounting Configuration Quick Start

Table 2-1 provides a quick overview of the steps required to create and configure authentication and accounting for the ACE. Each step includes the CLI command required to complete the task.

Table 2-1 Authentication and Accounting Configuration Quick Start

Task and Command Example
1. Configure the authentication and accounting service settings on the TACACS+, RADIUS, or LDAP server.

2. If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, log directly in to, or change to, the correct context.
`host1/Admin# changeto C1`
`host1/C1#`
The rest of the examples in this table use the Admin context, unless otherwise specified. For details on creating contexts and user accounts to provide access to the local database on the ACE for CLI access authentication, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

3. Enter configuration mode by entering **config**.
`host1/Admin# config`
`Enter configuration commands, one per line. End with CNTL/Z`
`host1/Admin(config)#`

Table 2-1 Authentication and Accounting Configuration Quick Start (continued)

Task and Command Example

4. Configure the individual AAA server parameters. For example, to configure RADIUS server authentication parameters, enter:
`host1/Admin(config)# radius-server host 192.168.2.3 key HostKey`
`host1/Admin(config)# radius-server host 192.168.2.3 key 7 secret_1256`
`host1/Admin(config)# radius-server host 192.168.2.3 auth-port 1645`
`host1/Admin(config)# radius-server host 192.168.2.3 acct-port 1646`
`host1/Admin(config)# radius-server host 192.168.2.3 authentication`
`host1/Admin(config)# radius-server host 192.168.2.3 accounting`
`host1/Admin(config)# radius-server host 192.168.2.3 timeout 25`
`host1/Admin(config)# radius-server host 192.168.2.3 retransmit 3`

5. Configure independent server groups of TACACS+, RADIUS, or LDAP servers. For example, to create a RADIUS server group, enter:
`host1/Admin(config)# aaa group server radius RAD_Server_Group1`
`host1/Admin(config-radius)# server 192.168.252.1`
`host1/Admin(config-radius)# server 192.168.252.2`
`host1/Admin(config-radius)# server 192.168.252.3`
`host1/Admin(config-radius)# deadtime 15`

6. Configure the authentication method used for login to the ACE CLI. `host1/Admin(config)# aaa authentication login console group RAD_Server_Group1 local none`
7. Configure the default accounting method. `host1/Admin(config)# aaa accounting default group RAD_Server_Group1 local`

8. (Optional) Save your configuration changes to flash memory.
`host1/Admin(config)# exit`
`host1/Admin# copy running-config startup-config`

Configuring the AAA Server

This section describes how to set up a TACACS+ or RADIUS server such as the Cisco Secure Access Control Server (ACS). It also covers general guidelines for setting up an LDAP directory server, such as OpenLDAP Software available from OpenLDAP Project. This section is intended as a guide to help ensure proper communication with the AAA server and an ACE operating as the AAA client.

For details on configuring the Cisco Secure ACS, OpenLDAP Software, or another AAA server, see the documentation that is provided with the software.

Configuring a TACACS+ Server

This section contains the following topics:

- Configuring Authentication Settings on the TACACS+ Server
- Configuring Accounting Settings on the TACACS+ Server
- Defining Private Attributes for Virtualization Support in a TACACS+ Server

Note For the ACE to properly perform user authentication using a TACACS+ server, the username and password must be identical on both the ACE and the TACACS+ server.

Configuring Authentication Settings on the TACACS+ Server

To configure the TACACS+ authentication settings on Cisco Secure ACS, perform the following steps:

Step 1 Go to the Network Configuration section of the Cisco Secure ACS HTML interface, and then go to the Add AAA Client page.

Step 2 Configure the following selections:

- AAA Client Hostname—Enter the name that you want assigned to the ACE.
- AAA Client IP Address—Enter the IP address of the Ethernet interface that will be used for communicating with the TACACS+ server.

- Key—Enter the shared secret that the ACE and Cisco Secure ACS use to authenticate transactions. You must specify the identical shared secret on both the Cisco Secure ACS and the ACE. The key is case sensitive.
- Authenticate Using—Choose **TACACS+ (Cisco IOS)**.

Note The TACACS+ (Cisco IOS) drop-down item is the title for the Cisco TACACS+ authentication function. The TACACS+ (Cisco IOS) selection activates the TACACS+ option when using Cisco Systems access servers, routers, and firewalls that support the TACACS+ authentication protocol. This includes support with an ACE as well.

Step 3 Click **Submit + Restart**.

Configuring Accounting Settings on the TACACS+ Server

To configure the TACACS+ accounting service for the Cisco Secure ACS, perform the following steps:

Step 1 In the System Configuration section of the Cisco Secure ACS interface, the Logging Configuration page, click **CSV TACACS+ Accounting**. The CSV TACACS+ Accounting File Configuration page appears.

Step 2 Confirm that the **Log to CSV TACACS+ Accounting report** check box is checked.

Step 3 Under Select Columns To Log, in the Attributes column, click the attribute that you want to log. Click **->** to move the attribute into the Logged Attributes column. Click **Up** or **Down** to move the column for this attribute to the desired position in the log. Repeat until all the desired attributes are in the desired positions in the Logged Attributes column.

Step 4 Click **Submit** when you finish moving the attributes into the Logged Attributes.

Defining Private Attributes for Virtualization Support in a TACACS+ Server

You can create the same username across contexts and associate it with a unique role in a context and multiple domains. Contexts can share a TACACS+ server, but the user must be authenticated for each context and must use the same password.

When a user attempts to log in to the ACE, the TACACS+ client on the ACE sends the username and password to the remote TACACS+ server for authentication. The TACACS+ server retrieves a user's profile as part of the authentication request. Once the user is successfully authenticated, the TACACS+ server returns a user profile to the TACACS+ client on the ACE with the authentication status. If the associated context of the user attempting to log in matches the contexts of the user profile obtained through the TACACS+ server, the TACACS+ client updates the user profile with the remote server user profile. If the contexts do not match, the user profile is updated with a default role (Network-Monitor) and a default domain (default-domain).

Configure the user profile on the TACACS+ server to run an Exec shell to configure a shell command authorization for the user. Define a custom attribute with a value string in the following format:

shell:<contextname>=<role> <domain1> <domain2>...<domainN>

or

shell:<contextname>*<role> <domain1> <domain2>...<domainN>

Note If you are using Cisco IOS command authorization, be sure to use an asterisk (*) rather than the equals sign (=) operator in the shell command string. The equals sign indicates that Cisco IOS software expects a required field to follow. Cisco IOS software does not recognize the role field, so using the equals sign in this case will cause Cisco IOS authorization to fail.

Note The user profile attribute serves an important configuration function for a TACACS+ server group. If the user profile attribute is not obtained from the server during authentication, or if the profile is obtained from the server but the context name(s) in the profile do not match the context in which the user is trying to log in, a default role (Network-Monitor) and a default domain (default-domain) are assigned to the user if the authentication is successful.

To configure the TACACS+ role and domain settings on Cisco Secure ACS, perform the following steps:

Step 1 Go to the Interface Configuration section of the Cisco Secure ACS HTML interface and access the TACACS+ (Cisco IOS) page. Perform the following actions:

a. Under the TACACS+ Services section of the page, the User column or the Group column depending on your configuration, check the **Shell (exec)** check box.

b. Under the Advanced Configuration Options section of the page, check the **Display a window for each service selected in which you can enter customized TACACS+ attributes** check box.

c. Click **Submit**.

Step 2 Go to the Advanced Options page of the Interface Configuration section of the Cisco Secure ACS HTML interface. Perform the following actions:

a. Check the **Per-user TACACS+/RADIUS Attributes** check box.

b. Click **Submit**.

Step 3 Go to the User Setup section of the Cisco Secure ACS HTML interface and double-click the name of an existing user that you want to define a user profile attribute for virtualization. The User Setup page appears.

Step 4 Under the TACACS+ Settings section of the page, configure the following settings:

- Check the **Shell (exec)** check box.
- Check the **Custom attributes** check box.
- In the text box under the Custom attributes, enter the user role and associated domain for a specific context in the following format:

```
shell:<contextname>=<role> <domain1> <domain2>...<domainN>
```

For example, to assign the selected user to the C1 context with the role ROLE1 and the domain DOMAIN1, enter **shell:C1=ROLE1 DOMAIN1**.

You can also substitute an asterisk (*) for the equals sign (=) as follows:

```
shell:<contextname>*<role> <domain1> <domain2>...<domainN>
```

Use the above shell string if you are also using Cisco IOS command authorization.

Step 5 Under the Checking This option Will PERMIT all UNKNOWN Services section of the page, check the **Default (Undefined) Services** check box to permit unknown services.

Step 6 Click **Submit** when you finish configuring the TACACS+ role and domain settings.

For example, if USER1 is assigned the role ADMIN and the domain MYDOMAIN1 (where shell:Admin=ADMIN MYDOMAIN1), then one of the following can occur:

- If USER1 logs in through the Admin context, that user is automatically assigned the Admin role and the MyDomain1 domain.
- If USER1 logs in through a different context, that user is automatically assigned the default role (Network-Monitor) and the default domain (default-domain). In this case, the user profile attribute is not obtained from the TACACS+ server during authentication.

Configuring a RADIUS Server

This section contains the following topics:

- Configuring Authentication Settings on the RADIUS Server
- Configuring Accounting Settings on the RADIUS Server
- Defining Private Attributes for Virtualization Support in a RADIUS Server

Configuring Authentication Settings on the RADIUS Server

To configure the RADIUS authentication settings on Cisco Secure ACS, perform the following steps:

Step 1 Go to the Network Configuration section of the Cisco Secure ACS HTML interface.

Note If you are using Network Device Groups (NDGs), you must also click the name of the NDG to which you want to add the AAA client entry.

Step 2 Under the AAA Clients table, choose **Add Entry**. The Add AAA Client page appears.

Step 3 Configure the entries on the Add AAA Client page as follows:

- AAA Client Hostname—Enter a name that you want assigned to the ACE.
- AAA Client IP Address—Enter the IP address of the Ethernet Management port or of an ACE circuit (depending on how the ACE is configured to communicate with the Cisco Secure ACS).
- Key—Enter the shared secret that the ACE and Cisco Secure ACS use to authenticate transactions. You must specify the identical shared secret on both the Cisco Secure ACS and the ACE. The key is case sensitive.
- Authenticate Using—Choose the AAA protocol. In the case of RADIUS, choose the vendor used for communication with the AAA client.

Step 4 Click **Submit + Restart**.

Cisco Secure ACS saves the AAA client entry and restarts its services, after which it will accept and process RADIUS requests from the ACE.

Configuring Accounting Settings on the RADIUS Server

To configure the RADIUS accounting settings on Cisco Secure ACS, perform the following steps:

Step 1 Choose **System Configuration > Logging > CSV RADIUS Accounting**. The CSV RADIUS Accounting File Configuration page appears.

Step 2 Confirm that the **Log to CSV RADIUS Accounting report** check box is checked.

Step 3 In the Select Columns To Log table, check that the RADIUS attributes that you want to see in the RADIUS accounting log appear in the Logged Attributes list. In addition to the standard RADIUS attributes, you will see several special logging attributes provided by Cisco Secure ACS, such as Real Name, ExtDB Info, and Logged Remotely. For more information about these attributes, see the user guide for your Cisco Secure ACS.

Step 4 (Optional) If you are using Cisco Secure ACS for Windows Server, you can specify log file management, which determines how large the RADIUS account files can be, how many are retained, how long they are retained, and where they are stored.

Note You can use Cisco Secure ACS to send accounting data to other AAA servers by configuring the AAA server entry in the Network Configuration section of the HTML interface. For details, see the applicable Cisco Secure ACS user guide.

Step 5 Click **Submit** when you finish moving the attributes into the Logged Attributes.

Cisco Secure ACS saves and implements the changes that you made to its RADIUS accounting configuration.

Defining Private Attributes for Virtualization Support in a RADIUS Server

You can create the same username across contexts and associate it with a unique role in a context and multiple domains. Contexts can share a RADIUS server, but the user must be authenticated for each context and must use the same password.

When a user attempts to log in to the ACE, the RADIUS client on the ACE sends the username and password to the remote RADIUS server for authentication. The RADIUS server retrieves a user's profile as part of the authentication request. Once the user is successfully authenticated, the RADIUS server returns a user profile to the RADIUS client on the ACE with the authentication status. If the associated context of the user attempting to log in matches the contexts of the user profile obtained through the RADIUS server, the RADIUS client updates the user profile with the remote server user profile. If the contexts do not match, the user profile is updated with the default role (Network-Monitor) and the default domain (default-domain).

Configure the user profile on the RADIUS server as a Vendor Specific Attribute with vendor Id Cisco (09) and subattribute type CiscoAVPair (type 01) with a value string in the following format:

shell:<contextname>=<role> <domain1> <domain2>...<domainN>

Note The user profile attribute serves an important configuration function for a RADIUS server group. If the user profile attribute is not obtained from the server during authentication, or if the profile is obtained from the server but the context name(s) in the profile do not match the context in which the user is trying to log in, a default role (Network-Monitor) and a default domain (default-domain) are assigned to the user if the authentication is successful.

To configure the RADIUS role and domain settings on Cisco Secure ACS, perform the following steps:

Step 1 Go to the User Setup section of the Cisco Secure ACS HTML interface and double-click the name of an existing user that you want to define a user profile attribute for virtualization. The User Setup page appears.

Step 2 Under the Cisco IOS/PIX RADIUS Attributes section of the page, configure the following settings:

- Check the **[009\001] cisco-av-pair** check box.
- In the text box below the [009\001] cisco-av-pair check box, enter the user role and associated domain for a specific context in the following format:

  ```
  shell:<contextname>=<role> <domain1> <domain2>...<domainN>
  ```

  For example, to assign the selected user to the C1 context with the role ROLE1 and the domain DOMAIN1, enter **shell:C1=ROLE1 DOMAIN1**.

Step 3 Click **Submit** when you finish configuring the RADIUS role and domain settings.

For example, if USER1 is assigned the role ADMIN and the domain MYDOMAIN1 (where shell:Admin=ADMIN MYDOMAIN1), then one of the following can occur:

- If USER1 logs in through the Admin context, that user is automatically assigned the Admin role and the MyDomain1 domain.
- If USER1 logs in through a different context, that user is automatically assigned the default role (Network-Monitor) and the default domain (default-domain). In this case, the user profile attribute is not obtained from the RADIUS server during authentication.

Configuring an LDAP Server

This section describes how to set up an LDAP directory server, such as the OpenLDAP and Microsoft Active Directory Servers. This section is intended as a general guide to help ensure proper communication with an LDAP server and an ACE operating as an LDAP client.

To configure the OpenLDAP directory server, perform the following steps:

Step 1 Edit the provided slapd.conf example (usually installed as /usr/local/etc/openldap/slapd.conf) to contain a BDB database definition, schema definition, rootDN, and root password.

Step 2 Add a private schema to include the definition of the private attributes (context ID and user profile) and private objectClass, or modify the existing object class. Include this schema in the slapd.conf.

Step 3 Start the LDAP server, slapd.

Note slapd is a standalone LDAP directory server that runs on many different platforms.

Step 4 Create the LDAP database; that is, create a file in LDIF format that contains the database. Ensure that the LDIF file (example.ldif) contains the following:

dn: dc=example,dc=com

objectClass: dcObject

objectClass: organization

dc: example

o: Example Corporation

description: The Example Corporation

dn: cn=Manager,dc=example,dc=com

objectclass: organizationalRole

cn: Manager

Step 5 Run **ldapadd** to insert these entries into your directory. For example:

```
ldapadd -x -D "cn=Manager,dc=example,dc=com" -w secret -f
example.ldif
```

Defining Private Attributes for Virtualization Support in an LDAP Server

The LDAP client on the ACE does not assume any specifics about the database structure maintained by the LDAP server. Instead, it assumes that the {userid, contextid} pair uniquely identifies an entry in the database and that this entry contains the user profile attribute. The LDAP client performs a search based on these two attributes using the search filter configured on the ACE. The LDAP server locates the correct user entry and the user profile attribute, which is part of that entry, and returns this information in the search response.

The LDAP client can operate in applications where virtualization is not a requirement. In this case, the username alone uniquely identifies the user entry. You configure the search filter to include only the $userid variable (no $contextid). You define these two private attributes from the ACE CLI by entering the **attribute user-profile** command (see the "Configuring the User Profile Attribute Type for an LDAP Server Group" section).

You define the user profile attribute value in the following format:

shell:<contextname>=<role> <domain1> <domain2>...<domainN>

Note The user profile attribute serves an important configuration function for an LDAP server group. If the user profile attribute is not obtained from the server during authentication, or if the profile is obtained from the server but the context name(s) in the profile do not match the context in which the user is trying to log in, a default role (Network-Monitor) and a default domain (default-domain) are assigned to the user if the authentication is successful.

When virtualization is a requirement, the LDAP server must have the contextid attributes defined in the schema. The user-profile attribute (the role-domain information) is required if you need to assign different roles and domains to different users. See the LDAP client documentation for information about how to extend the attributetype directive used by the slapd LDAP directory server.

To define private attributes for virtualization support in an LDAP server, perform the following steps:

Step 1 Add a private schema to include the definition of the private attributes (context ID and user profile) and the private objectClass. An example is as follows:

```
attributetype (2.5.4.55 NAME ( 'ctxid' 'contextid' )
         DESC 'virtual context name'
         SUP name )

attributetype ( 2.5.4.56 NAME ( 'usrprof' 'userprofile' )
         DESC 'user profile'
         SUP name )

objectclass ( 2.5.6.30 NAME 'ctxperson'
         DESC 'a person'
         SUP top STRUCTURAL
         MUST cn
         MAY  ( $ ctxid $ usrprof ) )
```

The example includes arbitrary OIDs. The OIDs that you define must not overlap with any existing OIDs in the LDAP server database.

Step 2 Include this private schema in the configuration, which would be sladp.conf in the case of OpenLDAP.

Step 3 Define the LDAP database in LDAP Data Interchange Format (LDIF) with entries that contain the context ID and the user profile. LDIF formats are defined in RFC 2849. An example is as follows:

```
dn: ctxid=admin,cn=john,ou=employees,dc=example,dc=com
objectClass: ctxperson
ctxid: admin
cn: john
usrprof: shell:Admin=ROLE-1 DOMAIN-1
userPassword: xxxxxxxx
```

Step 4 Start the LDAP server, which is slapd in the case of OpenLDAP.

The LDAP client and LDAP server initiate their interaction as follows:

- The LDAP client sends a bind request with the DN as the configured rootDN and the password as the configured root password for the server group.
- If the bind is successful, the LDAP client sends a search request that includes the following:
  - baseDN—Configured baseDN
  - scope—Subtree
  - search filter—Configured filter with the $userid and $contextid replaced with the actual username and context name, respectively
  - attributes—Configured attribute type for userprofile
- If the search is successful, the LDAP server extracts the matched DN and user profile attribute value from the search response where the matched DN is the DN for the user.
- Rebind as the user, which involves the LDAP client sending a bind request with the DN as the user DN and the password as the user password.
- If the bind is successful, the LDAP server returns an authentication PASS message and also includes the user profile attribute value in this message.
- The LDAP client sends an unbind request to the LDAP server.

Creating User Accounts

Every user associated with a virtual context has account information stored on the ACE. The authentication information, username, user password, password expiration date, and user role membership are all stored as part of each user's profile.

As the ACE global administrator, you can assign one user in each context as the context administrator. The context administrator can then log in to the context or contexts on the ACE for which he or she is responsible and create additional users.

If you do not assign a user role to a new user, the default user role is Network-Monitor. By default, the user is allowed to operate on all domains. For users that you create in the Admin context, the default scope of access is the entire device. For users that you create in other contexts, the default scope of access is the entire context. If you need to restrict a user's access, you must assign a role-domain pair.

Note the following when assigning a user for a context in the ACE:

- The same username can be created across contexts and can be associated with a unique role in a context and multiple domains. A user can have up to ten domains associated with a unique role in a context.
- Virtual contexts can share RADIUS, TACACS+, and LDAP servers; however, the user must be explicitly authenticated for each context and use the same password.
- All logged user accounting activities are distinguished in the ACE by the context in which a user has signed in.

For detailed information about creating contexts and user accounts to provide access to the local database on the ACE for CLI access authentication, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

Configuring the ACE as a Client of a RADIUS, TACACS+, or LDAP Server

You can specify one or more AAA server groups to identify the server and the remote authentication protocol, RADIUS, TACACS+, or LDAP. You can configure multiple AAA servers (of the same server type) for each server group.

For each AAA server, you can specify the following:

- The server IP address and port.
- Encryption key (shared secret) to authenticate communication between the ACE and AAA server (RADIUS and TACACS+ servers only).
- The number of times that the ACE retransmits an authentication request to a timed-out server before it declares the AAA server to be unresponsive and contacts the next AAA server in the group (RADIUS and TACACS+ servers only).
- The time interval that the ACE waits for a server to reply to an authentication request before retransmitting another request to the server.

- The time interval in which the ACE sends probes to a AAA server to verify whether the server is available and can receive authentication requests. The dead-time interval starts when the server does not respond to the number of authentication request transmissions.
- Independent server groups of TACACS+, RADIUS, or LDAP servers.

This section contains the following topics:

- Configuring RADIUS on the ACE
- Configuring TACACS+ on the ACE
- Configuring LDAP on the ACE
- Configuring AAA Server Groups

Configuring RADIUS on the ACE

The ACE supports the RADIUS protocol to communicate with a remote RADIUS server for authentication and accounting services. This section defines the configuration of the ACE to operate as a client of a RADIUS server.

This section contains the following topics:

- Setting the RADIUS Server Parameters
- Configuring the RADIUS NAS-IP-Address Attribute
- Setting the Global RADIUS Server Preshared Key
- Configuring the Global RADIUS Server Dead-Time Interval
- Setting the Global RADIUS Server Number of Retransmissions
- Setting the Global RADIUS Server Timeout Value

Setting the RADIUS Server Parameters

You can use the **radius-server host** command to specify the RADIUS server IP address, encrypted key, destination UDP port, and other options. You can also define multiple **radius-server host** commands to configure multiple RADIUS servers.

The syntax of this command is as follows:

radius-server host *ip_address* [**key** *shared_secret* [**0** *shared_secret* | **7** *shared_secret*]] [**auth-port** *port_number*] [**acct-port** *port_number*] [**authentication**] [**accounting**] [**timeout** *seconds*] [**retransmit** *count*]

The arguments, keywords, and options are as follows:

- *ip_address* —IP address for the RADIUS server. Enter the address in dotted-decimal IP notation (for example, 192.168.11.1).
- **key**—(Optional) Enables an authentication key for communication between the ACE and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. This key overrides the global setting of the **radius-server key** command. If you do not specify a key, the global value is used. RADIUS keys are always stored in encrypted form in persistent storage. The running configuration also displays keys in encrypted form.
- *shared_secret*—Key used to authenticate communication between the RADIUS client and server. The shared secret must match the one configured on the RADIUS server. Enter the shared secret as a case-sensitive string with no spaces with a maximum of 63 alphanumeric characters.
- **0**—(Optional) Configures a key specified in clear text (indicated by 0) to authenticate communication between the RADIUS client and server.
- **7**—(Optional) Configures a key specified in encrypted text (indicated by 7) to authenticate communication between the RADIUS client and server.
- **auth-port** *port_number*—(Optional) Specifies the UDP destination port for communicating authentication requests to the RADIUS server. By default, the RADIUS authentication port is 1812 (as defined in RFC 2138 and RFC 2139). If your RADIUS server uses a port other than 1812, use the **auth-port** keyword to configure the ACE for the appropriate port before you start the RADIUS service. The *port_number* argument is the RADIUS port number. Valid values are from 1 to 65535.
- **acct-port** *port_number*—(Optional) Specifies the UDP destination port for communicating accounting requests to the RADIUS server. By default, the RADIUS accounting port is 1813 (as defined in RFC 2138 and RFC 2139). If your RADIUS server uses a port other than 1813, use the **acct-port** keyword to configure the ACE for the appropriate port before you start the RADIUS service. The *port_number* argument is the RADIUS port number. Valid values are from 1 to 65535.

- **authentication**—(Optional) Specifies that the RADIUS server is used only for authentication.

  Note If you do not specify either the authentication or accounting options, the RADIUS server is used for both accounting and authentication.

- **accounting**—(Optional) Specifies that the RADIUS server is used only for accounting.

  Note If you do not specify either the authentication or accounting options, the RADIUS server is used for both accounting and authentication.

- **timeout** *seconds*—(Optional) By default, the ACE waits 1 second for the RADIUS server to reply to an authentication request before retransmitting an authentication request to the server. Use the **timeout** keyword to change the time interval that the ACE waits for the RADIUS server to reply to an authentication request before retransmitting a request. Valid entries are from 1 to 60 seconds. The default is 1 second. For the specified server, this command overrides the global setting that was assigned by using the **radius-server timeout** command.
- **retransmit** *count*—(Optional) By default, the ACE send a single authentication request to a timed-out RADIUS server before it stops transmission and attempts to contact the next identified AAA server. The retransmit option is the number of times that the ACE retransmits an authentication request to a timed-out RADIUS server before it declares the server to be unresponsive and contacts the next server in the group. If all servers in the group are unavailable for authentication and accounting, the ACE tries the local database if you configured it as a local fallback method using the **aaa authentication login** or the **aaa accounting default** command. Valid entries are from 1 to 5 attempts. The default is 1 attempt. For the specified server, this command overrides the global setting that was assigned by using the **radius-server retransmit** command.

For example, to configure RADIUS server authentication parameters, enter:

```
host1/Admin(config)# radius-server host 192.168.2.3 key HostKey
host1/Admin(config)# radius-server host 192.168.2.3 key 7 secret_1256
host1/Admin(config)# radius-server host 192.168.2.3 auth-port 1645
host1/Admin(config)# radius-server host 192.168.2.3 acct-port 1646
host1/Admin(config)# radius-server host 192.168.2.3 authentication
```

```
host1/Admin(config)# radius-server host 192.168.2.3 accounting
host1/Admin(config)# radius-server host 192.168.2.3 timeout 25
host1/Admin(config)# radius-server host 192.168.2.3 retransmit 3
```

To revert to a default RADIUS server authentication setting, enter:

```
host1/Admin(config)# no radius-server host 192.168.2.3 acct-port 1646
```

Configuring the RADIUS NAS-IP-Address Attribute

Typically, RADIUS servers check the source IP address in the IP header of the RADIUS packets to track the source of the RADIUS requests. Also, some servers use the NAS-IP-Address RADIUS attribute to identify the RADIUS clients that can expose your ACE internal private network interface IP addresses.

By default, the NAS-IP-Address is not configured. The ACE performs a route lookup on the RADIUS server IP address and uses the result. Use the **radius-server attribute nas-ipaddr** command to specify a RADIUS NAS-IP-Address attribute. This attribute allows you to configure an arbitrary IP address to be used as RADIUS attribute 4, NAS-IP-Address for each context. The **radius-server attribute nas-ipaddr** command allows the ACE to behave as a single RADIUS client from the perspective of the RADIUS server. The configured NAS-IP-Address is encapsulated in all outgoing RADIUS authentication request and accounting packets.

The syntax of this command is as follows:

radius-server attribute nas-ipaddr *nas_ip_address*

The *nas_ip_address* argument configures an IP address to be used as the RADIUS NAS-IP-Address, attribute 4.

For example, to specify a RADIUS NAS-IP-Address, enter:

```
host1/Admin(config)# radius-server attribute nas-ipaddr 192.168.1.1
```

To delete the RADIUS NAS-IP-Address and return to the default configuration, enter:

```
host1/Admin(config)# no radius-server attribute nas-ipaddr 192.168.1.1
```

Setting the Global RADIUS Server Preshared Key

You can globally configure an authentication key for communication between the ACE and the RADIUS daemon running on each RADIUS server by using the **radius-server key** command. The key is a text string that must match the encryption key used on the RADIUS server. RADIUS keys are always stored in encrypted form in persistent storage on the ACE. This global key is applied to those RADIUS servers in a named server group for which a shared secret is not individually configured by the **radius-server host** command.

The syntax of this command is as follows:

radius-server key {*shared_secret* | **0** *shared_secret* | **7** *shared_secret*}

The arguments and keywords are as follows:

- *shared_secret*—Key used to authenticate communication between the RADIUS client and server. The shared secret must match the one configured on the RADIUS server. Enter the shared secret as a case-sensitive string with no spaces and a maximum of 63 alphanumeric characters.
- **0**—Configures a key specified in clear text (indicated by 0) to authenticate communication between the RADIUS client and server.
- **7**—Configures a key specified in encrypted text (indicated by 7) to authenticate communication between the RADIUS client and server.

For example, to globally configure an authentication key to be sent in encrypted text (indicated by 7) to the RADIUS server, enter:

```
host1/Admin(config)# radius-server key 7 abe4DFeeweo00o
```

To delete the key, enter:

```
host1/Admin(config)# no radius-server key 7 abe4DFeeweo00o
```

Configuring the Global RADIUS Server Dead-Time Interval

During the dead-time interval, the ACE sends probe access-request packets to verify that the RADIUS server is available and can receive authentication requests. The dead-time interval starts when the server does not respond to the number of authentication request transmissions configured through either the **radius-server retransmit** command or the **radius-server host retransmit** command. When the server responds to a probe access-request packet, the ACE transmits the authentication request to the server.

Use the **radius-server deadtime** command to globally set the time interval in which the ACE verifies whether a nonresponsive server is operational.

This command causes the ACE to mark any RADIUS servers that fail to respond to authentication requests as dead. This action avoids the wait for the request to time out before trying the next configured server. The ACE skips a RADIUS server that is marked as dead by sending additional requests for the duration of the specified *minutes* argument.

The syntax of this command is as follows:

radius-server deadtime *minutes*

The *minutes* argument is the length of time that the ACE skips a nonresponsive RADIUS server for transaction requests. Valid entries are from 0 to 1440 minutes (24 hours). The default is 0.

For example, to globally configure a 15-minute dead-time interval for RADIUS servers that fail to respond to authentication requests, enter:

```
host1/Admin(config)# radius-server deadtime 15
```

To set the RADIUS server dead-time interval to 0, enter:

```
host1/Admin(config)# no radius-server deadtime 15
```

Setting the Global RADIUS Server Number of Retransmissions

By default, the ACE sends one authentication request to a RADIUS server before it declares the server to be unresponsive and contacts the next server in the group. Use the **radius-server retransmit** command to globally change the number of times that the ACE sends an authentication request to a RADIUS server. If all servers in the group are unavailable for authentication and accounting, the ACE tries the local database if you configured it as a local fallback method using the **aaa authentication login** or the **aaa accounting default** command. If you do not have a fallback method, the ACE continues to contact one of the AAA servers listed in the server group.

The ACE applies this global retransmission value to those RADIUS servers for which a value is not individually configured by the **radius-server host** command.

The syntax of this command is as follows:

radius-server retransmit *count*

The *count* argument is the number of times that the ACE attempts to connect to a RADIUS server before trying to contact the next available server. The range is from 1 to 5 times. The default is 1.

For example, to globally configure the number of retransmissions to 3, enter:

```
host1/Admin(config)# radius-server retransmit 3
```

To revert to the default of one transmission attempt, enter:

```
host1/Admin(config)# no radius-server retransmit 3
```

Setting the Global RADIUS Server Timeout Value

By default, the ACE waits 1 second for the RADIUS server to send a reply to an authentication request to an unresponsive server before retransmitting an authentication request to the server. Use the **radius-server timeout** command to globally change the time interval that the ACE waits for the RADIUS server to reply before retransmitting an authentication request to the RADIUS server. The ACE applies this global timeout value to those RADIUS servers for which a timeout value is not individually configured by the **radius-server host** command.

The syntax of this command is as follows:

radius-server timeout *seconds*

The *seconds* argument is the time in seconds between retransmissions to the RADIUS server. Valid entries are from 1 to 60 seconds. The default is 1 second.

For example, to globally configure the timeout value to 30 seconds, enter:

```
host1/Admin(config)# radius-server timeout 30
```

To revert to the default of 1 second between transmission attempts, enter:

```
host1/Admin(config)# no radius-server timeout 30
```

Configuring TACACS+ on the ACE

The ACE supports the TACACS+ protocol to communicate with a TACACS+ server for authentication and accounting services. This section defines the configuration of the ACE to operate as a client of a TACACS+ server.

This section contains the following topics:

- Setting the TACACS+ Server Parameters
- Setting the Global Preshared Key
- Setting the Global TACACS+ Server Dead-Time Interval
- Setting the Global TACACS+ Server Timeout Value

Setting the TACACS+ Server Parameters

You can use the **tacacs-server host** command to specify the TACACS+ server IP address, encrypted key, destination port, and other options. You can define multiple **tacacs-server host** commands to configure multiple TACACS+ servers.

The syntax of this command is as follows:

tacacs-server host *ip_address* [**key** [**0** | **7**] *shared_secret*]] [**port** *port_number*] [**timeout** *seconds*]

The arguments, keywords, and options are as follows:

- *ip_address* —IP address for the TACACS+ server. Enter the address in dotted-decimal IP notation (for example, 192.168.11.1).
- **key**—(Optional) Enables an authentication key for communication between the ACE and the daemon that runs on the TACACS+ server. The key is a text string that must match the encryption key used on the TACACS+ server. This key overrides the global setting of the **tacacs-server key** command. If you do not specify a key, the global value is used. TACACS+ keys are always stored in encrypted form in persistent storage. The running configuration also displays keys in encrypted form.
- **0**—(Optional) Configures a key specified in clear text (indicated by 0) to authenticate communication between the TACACS+ client and server.
- **7**—(Optional) Configures a key specified in encrypted text (indicated by 7) to authenticate communication between the TACACS+ client and server.
- *shared_secret*—Key used to authenticate communication between the TACACS+ client and server. The shared secret must match the one configured on the TACACS+ server. Enter the shared secret as a case-sensitive string with no spaces and a maximum of 63 alphanumeric characters. Alternatively, you can use spaces if you enclose the entire string in quotation marks.

- **port** *port_number*—(Optional) Specifies the TCP destination port for communicating authentication requests to the TACACS+ server. By default, the TACACS+ authentication port is 49 (as defined in RFC 1492). If your TACACS+ server uses a port other than 49, use the **port** keyword to configure the ACE for the appropriate port prior to starting the TACACS+ service. The *port_number* argument specifies the TACACS+ port number. Valid values are from 1 to 65535.
- **timeout** *seconds*—(Optional) By default, the ACE waits 1 second for the TACACS+ server to reply to an authentication request before it declares a timeout failure and attempts to contact the next server in the group. If all servers in the group are unavailable for authentication and accounting, the ACE tries the local database if you configured it as a local fallback method using the **aaa authentication login** or the **aaa accounting default** command. Use the **timeout** keyword to change the time interval that the ACE waits for the TACACS+ server to reply to an authentication request. Valid entries are from 1 to 60 seconds. The default is 1 second. For the specified server, this command overrides the global setting that was assigned by using the **tacacs-server timeout** command.

For example, to configure TACACS+ server authentication parameters, enter:

```
host1/Admin(config)# tacacs-server host 192.168.3.2 key HostKey
host1/Admin(config)# tacacs-server host 192.168.3.2 port 1645
host1/Admin(config)# tacacs-server host 192.168.3.2 timeout 5
```

To remove the TACACS+ server from the configuration, enter:

```
host1/Admin(config)# no tacacs-server host 192.168.3.2 key HostKey
```

Setting the Global Preshared Key

You can globally configure an authentication key for communication between the ACE and the TACACS+ daemon that runs on each TACACS+ server by using the **tacacs-server key** command. The key is a text string that must match the encryption key used on the TACACS+ server. TACACS+ keys are always stored in encrypted form in persistent storage on the ACE. This global key is applied to those TACACS+ servers in a named server group for which a shared secret is not individually configured by the **tacacs-server host** command.

The syntax of this command is as follows:

tacacs-server key [**0** | **7**] *shared_secret* [**timeout** *seconds*]

The arguments and keywords are as follows:

- *shared_secret*—Key used to authenticate communication between the TACACS+ client and server. The shared secret must match the one configured on the TACACS+ server. Enter the shared secret as a case-sensitive string with no spaces with a maximum of 63 alphanumeric characters or you can enter spaces if you enclose the entire key with quotation marks (for example, “my key”).
- **0**—(Optional) Configures a key specified in clear text (indicated by 0) to authenticate communication between the TACACS+ client and server.
- **7**—(Optional) Configures a key specified in encrypted text (indicated by 7) to authenticate communication between the TACACS+ client and server.
- **timeout** *seconds*—(Optional) Globally configures the time interval that the ACE waits for the TACACS+ server to reply before retransmitting an authentication request to the TACACS+ server. The *seconds* argument is the timeout value in seconds. Valid entries are from 1 to 60 seconds. By default, the ACE waits 1 second to receive a response from a TACACS+ server before it declares a timeout failure and attempts to contact the next server in the group. This option configures the same time interval as the **tacacs-server timeout** command.

For example, to globally configure an authentication key in encrypted text (indicated by 7) to authenticate communication between the TACACS+ client and server, enter:

```
host1/Admin(config)# tacacs-server key 7 abe4DFeeweo00o
```

To delete the key, enter:

```
host1/Admin(config)# no tacacs-server key 7 abe4DFeeweo00o
```

Setting the Global TACACS+ Server Dead-Time Interval

During the dead-time interval, the ACE sends probe access-request packets to verify that the TACACS+ server is available and can receive authentication requests. The dead-time interval starts when the server does not respond to an authentication request transmission. When the server responds to a probe access-request packet, the ACE retransmits the authentication request to the server.

Use the **tacacs-server deadtime** command to globally set the time interval in which the ACE verifies whether a nonresponsive server is operational.

This command causes the ACE to mark any TACACS+ servers that fail to respond to authentication requests as dead. This action avoids the wait for the request to time out before trying the next configured server. The ACE skips a TACACS+ server that is marked as dead by sending additional requests for the duration of the *minutes* argument.

The syntax of this command is as follows:

tacacs-server deadtime *minutes*

The *minutes* argument is the length of time that the ACE skips a nonresponsive TACACS+ server for transaction requests. Valid entries are from 0 to 1440 minutes (24 hours). The default is 0.

For example, to globally configure a 15-minute dead-time interval for TACACS+ servers that fail to respond to authentication requests, enter:

```
host1/Admin(config)# tacacs-server deadtime 15
```

To set the TACACS+ server dead-time interval to 0, enter:

```
host1/Admin(config)# no tacacs-server deadtime 15
```

Setting the Global TACACS+ Server Timeout Value

By default, the ACE waits 1 second to receive a response from a TACACS+ server before it declares a timeout failure and attempts to contact the next server in the group. Use the **tacacs-server timeout** command to globally change the time interval that the ACE waits for the TACACS+ server to reply before retransmitting an authentication request to the TACACS+ server. The ACE applies this global timeout value to those TACACS+ servers for which a timeout value is not individually configured by the **tacacs-server host** command.

The syntax of this command is as follows:

tacacs-server timeout *seconds*

The *seconds* argument is the timeout value in seconds. Valid entries are from 1 to 60 seconds. The default is 1 second.

For example, to globally configure the timeout value to 30 seconds, enter:

```
host1/Admin(config)# tacacs-server timeout 30
```

To revert to the default of 1 second between transmission attempts, enter:

```
host1/Admin(config)# no tacacs-server timeout 30
```

Configuring LDAP on the ACE

The ACE supports the LDAP protocol to communicate with a remote LDAP directory server for authentication services. This section defines the configuration of the ACE to operate as a client of an LDAP server.

This section contains the following topics:

- Setting the LDAP Server Parameters
- Setting the Global LDAP Server Port Setting
- Setting the Global LDAP Server Timeout Value

Setting the LDAP Server Parameters

You can use the **ldap-server host** command to specify the LDAP server hostname or IP address, destination port, and other options. You can define multiple **ldap-server host** commands to configure multiple LDAP servers.

The syntax of this command is as follows:

ldap-server host *ip_address* [**port** *port_number*] [**timeout** *seconds*] [**rootDN** *"DN_string"* [**password** *bind_password*]]

The arguments, keywords, and options are as follows:

- *ip_address* —IP address for the LDAP server. Enter the address in dotted-decimal IP notation (for example, 192.168.11.1).
- **port** *port_number*—(Optional) Specifies the TCP destination port for communicating authentication requests to the LDAP directory server. By default, the LDAP server port is 389. If your LDAP server uses a port other than 389, use the **port** keyword to configure the ACE for the appropriate port before you start the LDAP service. The *port_number* argument is the LDAP port number. Valid values are from 1 to 65535. For the specified server, this command overrides the global setting that was assigned by using the **ldap-server port** command.
- **timeout** *seconds*—(Optional) Specifies the time in seconds to wait for a response from the LDAP server before the ACE can declare a timeout failure with the LDAP server. By default, the ACE waits 5 seconds for the LDAP server to reply to an authentication request before the ACE declares a timeout failure and attempts to contact the next server in the group. Use the **timeout**

keyword to change the time interval that the ACE waits for the LDAP server to reply to an authentication request. Valid entries are from 1 to 60 seconds. The default is 5 seconds. For the specified server, this command overrides the global setting that was assigned by using the **ldap-server timeout** command.

- **rootDN** *"DN_string"*—(Optional) Defines the distinguished name (DN) for a user who is unrestricted by access controls or administrative limit parameters to perform operations on the LDAP server directory. The rootDN user is the root user for the LDAP server database. Enter a quoted string that has a maximum of 63 alphanumeric characters. The default is an empty string.
- **password** *bind_password*—(Optional) Defines the bind password (rootpw) applied to the rootDN of the LDAP server directory. Enter an unquoted string that has a maximum of 63 alphanumeric characters. The default is an empty string.

For example, to configure LDAP server authentication parameters, enter:

```
host1/Admin(config)# ldap-server host 192.168.2.3 port 2003
host1/Admin(config)# ldap-server host 192.168.2.3 timeout 60
host1/Admin(config)# ldap-server host 192.168.2.3 rootDN
"cn=manager,dc=cisco,dc=com" password lab
```

To remove the LDAP server authentication setting, enter:

```
host1/Admin(config)# no ldap-server host 192.168.2.3
```

Setting the Global LDAP Server Port Setting

By default, the TCP destination port for communicating authentication requests to the LDAP directory server is 389. If your LDAP server uses a port other than port 389, use the **ldap-server port** command to globally configure the ACE for the appropriate port before you start the LDAP service. This global port setting will be applied to those LDAP servers for which a TCP port value is not individually configured by the **ldap-server host** command.

The syntax of this command is as follows:

ldap-server port *port_number*

The *port_number* argument is the destination port to the LDAP server. Valid values are from 1 to 65535. The default is TCP port 389.

For example, to globally configure the TCP port, enter:

```
host1/Admin(config)# ldap-server port 2003
```

To revert to the default of TCP port 389, enter:

```
host1/Admin(config)# no ldap-server port 2003
```

Setting the Global LDAP Server Timeout Value

By default, the ACE waits 5 seconds to receive a response from an LDAP server before it declares a timeout failure and attempts to contact the next server in the group. Use the **ldap-server timeout** command to globally change the time interval that the ACE waits for the LDAP server to reply to a response before it declares a timeout failure. The ACE applies this global timeout value to those LDAP servers for which a timeout value is not individually configured by the **ldap-server host** command.

The syntax of this command is as follows:

ldap-server timeout *seconds*

The *seconds* argument is the timeout value in seconds. Valid entries are from 1 to 60 seconds. The default is 5 seconds.

For example, to globally configure the timeout value to 30 seconds, enter:

```
host1/Admin(config)# ldap-server timeout 30
```

To change to the default of 5 seconds between transmission attempts, enter:

```
host1/Admin(config)# no ldap-server timeout 30
```

Configuring AAA Server Groups

This section contains the following topics:

- Creating a TACACS+, RADIUS, or LDAP Server Group
- Setting the Dead-Time Interval for a TACACS+ Server Group
- Setting the Dead-Time Interval for a RADIUS Server Group
- Configuring the User Profile Attribute Type for an LDAP Server Group

- Configuring the Base DN for an LDAP Server Group
- Configuring the Search Filter for an LDAP Server Group

Creating a TACACS+, RADIUS, or LDAP Server Group

A server group is a list of server hosts of a particular type. The ACE allows you to configure multiple TACACS+, RADIUS, and LDAP servers as a named server group. You group the different AAA server hosts into distinct lists. The ACE searches for the server hosts in the order in which you specify them within a group.

Use the **aaa group server** command to configure independent server groups of TACACS+, RADIUS, or LDAP servers. You can configure server groups at any time, but they only take effect when you apply them to the AAA service using the **aaa authentication login** or the **aaa accounting default** commands.

You can configure a maximum of 10 server groups for each context in the ACE.

The ACE attempts to contact the first server listed in the server group for user authentication and accounting. If that server is unavailable, the ACE attempts to contact the next configured server listed in the group. If all servers in the group are unavailable, the ACE then tries the servers in the next configured server group. The ACE repeats this process until the authentication request can be handled by an AAA server. If the specified AAA servers in a server group are unavailable, and you specify local authentication as the fallback method (as specified in the **aaa authentication login** command), the ACE attempts to authenticate the user against the local database on the ACE. If you do not have a fallback method, the ACE continues to contact one of the AAA servers listed in the server group.

The syntax of this command is as follows:

aaa group server {**ldap** | **radius** | **tacacs+**} *group_name*

The arguments and keywords are as follows:

- **ldap**—Specifies an LDAP directory server group.
- **radius**—Specifies a RADIUS server group.
- **tacacs+**—Specifies a TACACS+ server group.
- *group_name*—Group of servers. The server group name is a maximum of 64 alphanumeric characters with no spaces.

The CLI displays the TACACS+, RADIUS, or LDAP server configuration mode where you identify the name of one or more previously configured servers that you want added to the server group.

The syntax of this server configuration mode command is as follows:

server *ip_address*

The *ip_address* argument is the IP address for an existing RADIUS, TACACS+, or LDAP server that you want to add to the server group. Enter the address in dotted-decimal IP notation (for example, 192.168.11.1). You can add multiple servers to the server group by entering multiple **server** commands while in server configuration mode. The same server can belong to multiple server groups.

For example, to create a RADIUS server group, enter:

```
host1/Admin(config)# aaa group server radius RAD_Server_Group1
host1/Admin(config-radius)# server 192.168.252.1
host1/Admin(config-radius)# server 192.168.252.2
host1/Admin(config-radius)# server 192.168.252.3
```

To remove a server from a server group, enter:

```
host1/Admin(config-radius)# no server 192.168.252.3
```

To remove a server group, enter:

```
host1/Admin(config)# no aaa group server radius RAD_Server_Group1
```

For the TACACS+, RADIUS, and LDAP server groups, you can also configure the following parameters:

- For a TACACS+ server group, you can specify a dead-time interval for the server group. See the "Setting the Dead-Time Interval for a TACACS+ Server Group" section.
- For a RADIUS server group, you can specify a dead-time interval for the server group. See the "Setting the Dead-Time Interval for a RADIUS Server Group" section.
- For an LDAP server group, you may specify the following parameters:
  - User profile attribute—See the "Configuring the User Profile Attribute Type for an LDAP Server Group" section.
  - Base DN—See the "Configuring the Base DN for an LDAP Server Group" section.

- LDAP search filter—See the "Configuring the Search Filter for an LDAP Server Group" section.

Setting the Dead-Time Interval for a TACACS+ Server Group

For a TACACS+ server group, you can specify a dead-time interval for the server group. During the dead-time interval, the ACE sends probe access-request packets to verify that the TACACS+ server is available and can receive authentication requests. The dead-time interval starts when the server does not respond to an authentication request transmission. When the server responds to a probe access-request packet, the ACE retransmits the authentication request to the server.

Use the **deadtime** command to globally set the time interval in which the ACE verifies whether a nonresponsive server group is operational.

This command causes the ACE to mark any TACACS+ servers that fail to respond to authentication requests as dead. This action avoids the wait for the request to time out before trying the next configured server. The ACE skips a TACACS+ server that is marked as dead by sending additional requests for the duration of the *minutes* argument.

The syntax of this command is as follows:

deadtime *minutes*

The *minutes* argument is the length of time that the ACE skips a nonresponsive TACACS+ server for transaction requests. Valid entries are from 0 to 1440 minutes (24 hours). The default is 0.

For example, to globally configure a 15-minute dead-time interval for TACACS+ servers that fail to respond to authentication requests, enter:

```
host1/Admin(config-tacacs)# deadtime 15
```

To reset the RADIUS server dead-time interval to 0, enter:

```
host1/Admin(config-tacacs)# no deadtime 15
```

Setting the Dead-Time Interval for a RADIUS Server Group

For a RADIUS server group, you can specify a dead-time interval for the server group. During the dead-time interval, the ACE sends probe access-request packets to verify that the RADIUS server is available and can receive authentication requests. The dead-time interval starts when the server does not respond to an authentication request transmissions. When the server responds to a probe access-request packet, the ACE retransmits the authentication request to the server.

Use the **deadtime** command to globally set the time interval in which the ACE verifies whether a nonresponsive server group is operational.

This command causes the ACE to mark any RADIUS servers that fail to respond to authentication requests as dead. This action avoids the wait for the request to time out before trying the next configured server. The ACE skips a RADIUS server that is marked as dead by sending additional requests for the duration of the *minutes* argument.

The syntax of this command is as follows:

deadtime *minutes*

The *minutes* argument is the length of time that the ACE skips a nonresponsive RADIUS server for transaction requests. Valid entries are from 0 to 1440 minutes (24 hours). The default is 0.

For example, to globally configure a 15-minute dead-time interval for RADIUS servers that fail to respond to authentication requests, enter:

```
host1/Admin(config-radius)# deadtime 15
```

To reset the RADIUS server dead-time interval to 0, enter:

```
host1/Admin(config-radius)# no deadtime 15
```

Configuring the User Profile Attribute Type for an LDAP Server Group

An LDAP server retrieves a user's profile as part of the search request. During a search request, the LDAP client requests the user profile attribute from the LDAP server by including this attribute type (the configured string) in the search request. The search request must match the attribute type used by the LDAP server to properly identify the user profile attribute, as defined in private schema on the LDAP server. The LDAP server uses the search filter to locate the user profile

entry in its database. When the LDAP server finds the entry, it replies with a search response in which it includes the value of the user profile attribute that was stored in that entry. This value contains the role and domain pair of the user for that context.

You define the user profile attribute value in the following format:

shell:<contextname>=<role> <domain1> <domain2>...<domainN>

Note The user profile attribute serves an important configuration function for an LDAP server group. If the user profile attribute is not obtained from the server during authentication, or if the profile is obtained from the server but the context name(s) in the profile do not match the context in which the user is trying to log in, a default role (Network-Monitor) and a default domain (default-domain) are assigned to the user if the authentication is successful.

This attribute type is used for the user profile attribute. Since this attribute type is private, the LDAP server database should use the same attribute type for the user profile. The LDAP client (the ACE) sends the search request with this attribute type as the attribute it wants to download. If the lookup was successful, the search response contains this attribute value. The attribute value must contain a string that represents the user role and domain pair for this particular context.

Use the **attribute user-profile** command to specify which user profile attribute to use by the LDAP server.

You can configure the LDAP user profile attribute at the subconfiguration level for the LDAP server group (created as described in the "Configuring AAA Server Groups" section).

The syntax of this command is as follows:

attribute user-profile *text*

The *text* argument is the user profile. The user profile is an unquoted text string of a maximum of 63 alphanumeric characters without spaces.

For example, to configure an LDAP user profile attribute, enter:

```
host1/Admin(config-ldap)# attribute user-profile usrprof
```

To delete the user profile attribute, enter:

```
host1/Admin(config-ldap)# no attribute user-profile usrprof
```

Configuring the Base DN for an LDAP Server Group

When you create an LDAP server group, the top level of the LDAP directory tree is the base, referred to as the base DN. The base DN is used to perform the search operation in the LDAP server directory. A base DN can take a form such as "dc=your,dc=domain", where the base DN uses the DNS domain name as its basis and is split into the domain components. Use the base-DN server group command to configure the base DN that you want to use to perform search operations in the LDAP directory tree.

Note The base DN is a mandatory configuration for an LDAP server group. Without this setting, a user cannot be authenticated.

You configure the base DN at the submode for the LDAP server group (created as described in the "Configuring AAA Server Groups" section).

The syntax of this command is as follows:

base-DN *text*

The *text* argument is the distinguished name of the search base. The base DN name is a quoted text string of a maximum of 63 alphanumeric characters without spaces.

For example, to configure the base DN, enter:

```
host1/Admin(config)# aaa group server ldap LDAP_Server_Group1
host1/Admin(config-ldap)# base-DN "dc=sns,dc=cisco,dc=com"
```

To delete the configured base DN, enter:

```
host1/Admin(config-ldap)# no base-DN "dc=sns,dc=cisco,dc=com"
```

Configuring the Search Filter for an LDAP Server Group

For an LDAP server group, the ACE transmits a search filter to the LDAP server to look up a user in the database. Search filters enable you to define search criteria and provide more efficient and effective searches. The search filter is used in the search request sent by the LDAP client to the server to locate the user's node in the DIT. Use the **filter search-user** command to configure the exact filter to use. The $user and $contextid are substituted with actual values when sending the request.

The search filter should follow the format defined in RFC 2254. The LDAP client sends the search request with the configured search filter after replacing the $userid and $contextid with the userid that the client is trying to authenticate and the associated virtual context name.The ACE allows $userid and $contextid to be used as placeholders for the user ID and the context name.

Note The search filter is a mandatory configuration for an LDAP server group. Without this setting, a user cannot be authenticated.

You configure the LDAP search filter at the subconfiguration level for the LDAP server group (created as described in the "Configuring AAA Server Groups" section).

The syntax of this command is as follows:

filter search-user *text*

The *text* argument is the search request. The search filter is a quoted text string of a maximum of 63 alphanumeric characters without spaces.

For example, to configure a search request, enter:

```
host1/Admin(config)# aaa group server ldap LDAP_Server_Group1
host1/Admin(config-ldap)# filter search-user "(&(objectclass=person)
(&(cn=$userid)(cid=$contextid)))"
```

To delete the search request, enter:

```
host1/Admin(config-ldap)# no filter search-user
"(&(objectclass=person)(&(cn=$userid)(cid=$contextid)))"
```

Defining the Login Authentication Method

Authentication is the process of verifying the identity of the person attempting to log in to the ACE CLI by console port or by a Telnet or SSH session. This identity verification is based on the username and password combination provided by the person attempting to access the ACE.

The ACE supports local authentication using the lookup database on the ACE or remote authentication using one or more TACACS+, RADIUS, or LDAP servers. You can specify the local database on the ACE as the fallback authentication method in case the configured AAA servers fail to respond to the authentication request.

The default login method of user authentication is by console port or by a Telnet or SSH session. You can override the default login authentication method and specify authentication through only the console port.

To configure the authentication method used for login to the ACE CLI, use the **aaa authentication login** command in configuration mode.

The syntax of this command is as follows:

aaa authentication login {{**console** | **default**} {{**group** *group_name*} {**local**} {**none**}}} | **error-enable**

The arguments, keywords, and options are as follows:

- **console—**Specifies the console port login authentication method, identified by the specified server group.
- **default**—Specifies the default login authentication method (console port or by a Telnet or SSH session), identified by the specified server group.
- **group** *group_name*—Associates the login authentication process with a TACACS+, RADIUS, or LDAP server defined through the **aaa group server** command. The server group name is a maximum of 64 alphanumeric characters with no spaces.
- **local**—Specifies to use the local database on the ACE as the login authentication method. If the server does not respond, then the local database is used as the fallback authentication method.
- **none**—Specifies that the ACE does not perform password verification. If you configure this option, users can log in to the ACE without providing a valid password. Only a user with an Admin role is allowed to specify the **none** option.

Caution Use this option with care. If you specify **none**, any user will be able to access the ACE at any time.

- **error-enable**—Enables the display of the login error message in instances where the remote AAA servers fail to respond. To view the current display status, use the **show aaa authentication login error-enable** command. When a user attempts to log in, and the remote AAA servers do not respond to the authentication request, the ACE processes the login sequence by switching to a local user database. If you activate the error-enabled feature, the following message appears on the user's terminal:

  ```
  Remote AAA servers unreachable; local authentication done.
  ```

For example, to enable console authentication using the TacServers server group, followed by local login as the fallback method, enter:

```
host1/Admin(config)# aaa authentication login console group TacServers
local
```

Password verification remains enabled for login authentication.

For example, to turn off password validation, enter:

```
host1/Admin(config)# aaa authentication login console group TacServers
local none
```

For example, to revert to the local authentication method, enter:

```
host1/Admin(config)# no aaa authentication login console group
TacServers local none
```

Defining the Default Accounting Method

Accounting refers to the log information that is maintained for each user's management session with an ACE. This information may be used to generate reports for troubleshooting and auditing purposes. Accounting can be implemented locally on the ACE or remotely using a RADIUS or TACACS+ server.

Use the **aaa accounting default** command to configure the default accounting method. You specify either a previously created AAA server group that identifies separate groups of TACACS+ or RADIUS servers or the local database on the ACE.

The syntax of this command is as follows:

aaa accounting default {**group** *group_name*} {**local**} {**none**}

The arguments and keywords are as follows:

- **group** *group_name*—Associates the accounting method with a TACACS+ or RADIUS server defined previously through the **aaa group server** command. The server group name is a maximum of 64 alphanumeric characters with no spaces.
- **local**—Specifies to use the local database on the ACE as the accounting method.
- **none**—Specifies that the ACE does not perform password verification, which disables password verification. If you configure this option, users can log in without providing a valid password.

Caution Use this option with care. If configured, any user will be able to access the ACE at any time.

For example, to enable user accounting to be performed using remote TACACS+ servers, followed by local login as the fallback method, enter:

```
host1/Admin(config-context)# aaa accounting default group TacServers
local
```

To revert to the default local accounting method, enter:

```
host1/Admin(config-context)# no aaa accounting default group
TacServers local
```

Viewing AAA Status and Statistics

This section contains the following topics:

- Displaying AAA Groups
- Displaying RADIUS Server Configuration Information
- Displaying TACACS+ Server Configuration Information
- Displaying LDAP Server Configuration Information
- Displaying Accounting Configuration Information
- Displaying Accounting Log Information
- Displaying Authentication Configuration Information

Displaying AAA Groups

You can display the configured server groups by using the **show aaa groups** command. The syntax of this command is as follows:

show aaa groups

For example, to display configured server groups, enter:

```
host1/Admin# show aaa groups
    TACACS:
        TACACS_group1
    RADIUS:
        RAD_group1
    LDAP:
        LDAP_group2
```

Displaying RADIUS Server Configuration Information

You can display the configured RADIUS server and group parameters by using the **show radius-server** command.

The syntax of this command is as follows:

show radius-server [**groups** | **sorted**]

The optional keywords are as follows:

- **groups**—(Optional) Displays configured RADIUS server group information.
- **sorted**—(Optional) Displays RADIUS server information sorted by name.

For example, to display configured RADIUS server parameters, enter:

```
host1/Admin# show radius-server
    retransmission count:1
    timeout value:1
    deadtime value:20
    total number of servers:2

    following RADIUS servers are configured:
            192.168.34.45:
                    available for authentication on port:1812
                    available for accounting on port:1813
            192.168.2.3:
                    available for authentication on port:1812
```

```
                    available for accounting on port:1813
                    RADIUS shared secret:********
```

For example, to display the configured RADIUS server groups, enter:

```
host1/Admin# show radius-server groups
    total number of groups:2

    following RADIUS server groups are configured:
            group radius:
                    server: all configured radius servers
            group RAD_Server_Group:
                    deadtime is 0
```

For example, to display the sorted RADIUS server groups, enter:

```
host1/Admin# show radius-server sorted
    retransmission count:1
    timeout value:1
    deadtime value:20
    total number of servers:2

    following RADIUS servers are configured:
            192.168.34.45:
                    available for authentication on port:1812
                    available for accounting on port:1813
            192.168.2.3:
                    available for authentication on port:1812
                    available for accounting on port:1813
                    RADIUS shared secret:********
```

Displaying TACACS+ Server Configuration Information

You can display the configured TACACS+ server and group parameters by using the **show tacacs-server** command.

The syntax of this command is as follows:

show tacacs-server [**groups** | **sorted**]

The optional keywords are as follows:

- **groups**—(Optional) Displays configured TACACS+ server group information.
- **sorted**—(Optional) Displays TACACS+ server information sorted by name.

For example, to display the configured TACACS+ server parameters, enter:

```
host1/Admin# show tacacs-server
    Global TACACS+ shared secret:tacacsPword
    timeout value:30
    total number of servers:3

    following TACACS+ servers are configured:
    192.168.58.91:
    available on port:2
    cisco.com:
    available on port:49
    192.168.22.95:
    available on port:49
    TACACS+ shared secret:MyKey
```

For example, to display the configured TACACS+ server groups, enter:

```
host1/Admin# show tacacs-server groups
    total number of groups:1

    following TACACS+ server groups are configured:
    group TacServers:
    server 192.168.58.91 on port 2
```

For example, to display the sorted TACACS+ servers, enter:

```
host1/Admin# show tacacs-server sorted
    timeout value:1
    total number of servers:1
```

Displaying LDAP Server Configuration Information

You can display the configured LDAP server and server group parameters by using the **show ldap-server** command.

The syntax of this command is as follows:

show ldap-server [**groups**]

The optional **groups** keyword displays configured LDAP server group information.

To display configured LDAP server parameters, enter:

```
host1/Admin# show ldap
    timeout : 5
            port : 389
```

```
    total number of servers : 1
```

To display the configured LDAP server groups, enter:

```
host1/Admin# show ldap-server groups
    total number of groups: 1

    following LDAP server groups are configured:
        group LDAP_Server_Group1:
            baseDN: "dc=sns,dc=cisco,dc=com"
            user profile attribute: usrprof
            search filter: "(&(objectclass=person)
    (&(cn=$userid)(cid=$contextid)))"
```

Displaying Accounting Configuration Information

You can display accounting configuration information for the ACE by using the **show aaa accounting** command.

The syntax of this command is as follows:

show aaa accounting

For example, to display accounting configuration information, enter:

```
host1/Admin# show aaa accounting
    default: local
```

Displaying Accounting Log Information

You can display accounting log information for the ACE by using the **show accounting log** command.

The syntax of this command is as follows:

show accounting log [*size*] [**all**]

The argument and option are as follows:

- *size*—(Optional) The size of the local accounting log file to display in bytes from 0 to 250000. The default is 250000 bytes.
- **all**—(Optional) Displays the accounting logs of all contexts in the ACE. This option is available only in the Admin context.

For example, to display accounting log information, enter:

```
host1/Admin# show accounting log
    Sat Jan  1 00:02:55 2000:start:/dev/ttyS00_946684975:admin:
    Sat Nov  5 00:20:04 2005:update:/dev/ttyS00_946684975:admin:0:ft
    interface vlan
    50
    Sat Nov  5 00:20:05 2005:update:/dev/ttyS00_946684975:admin:1:ip
    address 12.1.1.
    2 255.255.255.0
    Sat Nov  5 00:20:05 2005:update:/dev/ttyS00_946684975:admin:1:peer
    ip 12.1.1.1 2
    55.255.255.0
    Sat Nov  5 00:20:05 2005:update:/dev/ttyS00_946684975:admin:1:no
    shutdown
    Sat Nov  5 00:20:12 2005:update:/dev/ttyS00_946684975:admin:0:ft
    peer 1
    Sat Nov  5 00:20:12
    2005:update:/dev/ttyS00_946684975:admin:0:ft-interface vlan
    50
    Sat Nov  5 00:20:41 2005:update:/dev/ttyS00_946684975:admin:0:log
    console 6
    Sat Nov  5 00:20:58 2005:update:/dev/ttyS00_946684975:admin:0:ft
    group 1
    Sat Nov  5 00:20:58 2005:update:/dev/ttyS00_946684975:admin:0:peer
    1
    Sat Nov  5 00:20:58
    2005:update:/dev/ttyS00_946684975:admin:0:priority 50
    Sat Nov  5 00:20:58
    2005:update:/dev/ttyS00_946684975:admin:0:associate-context
    Admin
    Sat Nov  5 00:20:58
    2005:update:/dev/ttyS00_946684975:admin:0:inservice
    .
    .
```

The ACE now includes the following configuration mode commands in the accounting logs:

- [**no**] **ldap-server host** *ip_address* [**port** *port_number*] [**timeout** *seconds*] [**rootDN** "*DN_string*" [**password** *bind_password*]]
- [**no**] **radius-server key** [**0** | **7**] *shared_secret*
- [**no**] **radius-server host** *ip_address* **key** [**0** | **7**] *shared_secret*
- [**no**] **snmp-server community** *community_name*
- [**no**] **snmp-server host** *ip_address* [**inform** | **traps**] [**version** {**1** | **2c**} | {**3** {**auth** | **noauth** | **priv**}}] *community_ string_or_username*

- [**no**] **snmp-server user** *user_name* [*group_name*] [**auth** {**md5** | **sha**} **password1** [**priv** {*password2* | **aes-128** *password2*}] [**localizedkey**]]
- [**no**] **tacacs-server host** *ip_address* **key** [**0** | **7**] *shared_secret*
- [**no**] **tacacs-server key** [**0** | **7**] *shared_secret*
- [**no**] **username** *name1* [**password** [**0** | **5**] {*password*}]

Previously, the ACE omitted these commands from the logs because they contain sensitive information, such as a community name, shared secret, username, or password.

With this behavior change, when the ACE includes any of these commands in the log, it masks the sensitive information with five stars. For example, when you enter the **snmp-server community** *community_name* command, the ACE logs the following:

```
snmp-server community *****
```

Note The ACE logs the sensitive information for the following commands in plain text and does not mask it:

- The **backup pass-phrase** command in Exec mode
- The **ip address** command in KAL-AP UDP configuration mode
- The **credentials** command in probe RADIUS configuration mode

The ACE does not save the following CLI commands in the accounting log because the commands are handled by VSH and they do not go through the accounting framework:

- **clear screen**
- **config terminal**
- (**no**) **debug**
- **end**
- **exit**
- **show debug**
- **show terminal**
- **terminal** (**no**)

- **(no) username**
- **xml-show on**
- **xml-show off**
- **xml-show status**

Displaying Authentication Configuration Information

You can display authentication configuration information for the ACE by using the **show aaa authentication** command.

The syntax of this command is as follows:

show aaa authentication [**login error-enable**]

The optional **login error-enable** keyword allows you to view the current display status of the login error message.

For example, to display the configured authentication parameters, enter:

```
host1/Admin# show aaa authentication
    default: group TacServers local none
    console: local

host1/Admin# show aaa authentication login error-enable
    enabled
```